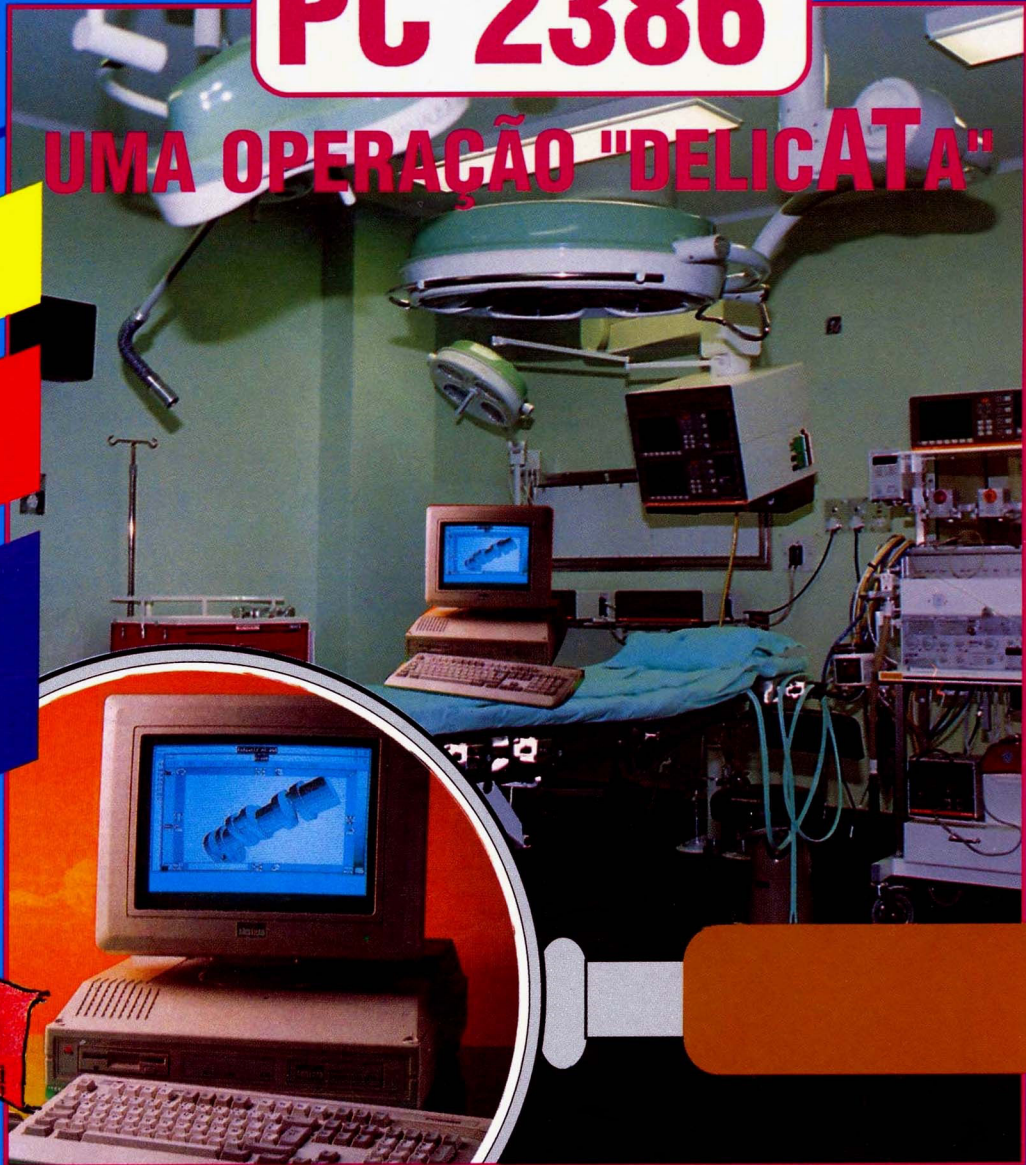
AMSTRAD MAGAZINE

REVISTA MENSAL Nº 11 ANO 1 ABRIL 1989 350$00

REVISTA DOS UTILIZADORES AMSTRAD

PC 2386

UMA OPERAÇÃO "DELICATA"

MEMÓRIA?

DIGITALIZAÇÃO DE SONS PARA O CPC

MISSIONÁRIOS, CANIBAIS E MUITO, MUITO MAIS

WINDOWS NÃO TINHA... UM TECLADO PORTUGUÊS

as melhores marcas!
os melhores precos!
as melhores condicões de pagamento!

COMPUTADORES OLIVETTI PC1

PESSOAL ESPECIALIZADO

COMPUTADORES AMSTRAD

EXCEL é a sua cadeia de lojas de electrodomésticos.Onde Você pode adquirir as melhores marcas e beneficiar das maiores facilidades de pagamento. Visite as nossas secções de computadores. Dispomos do mais elevado profissionalismo para o atender. Venha conhecer-nos!

LOJAS EXCEL

- LISBOA — C.C. Amoreiras, Loja 2102 ☎ 69 33 58
  — C.C. Pingo Doce — Alcântara
  — Alameda das Linhas de Torres, 257 (Lumiar) — ☎ 758 82 68
- ALMADA · C.C. Pão de Açúcar ☎ 275 21 83
- LINDA-A-VELHA — C.C. Pingo Doce ☎ 97 75 83
- VENDA NOVA — C.C. Pingo Doce ☎ 97 75 83
- ALFRAGIDE — C.C. Pão de Açúcar (Jumbo) ☎ 418 80 89
- LOURES — C.C. Modelo, lojas 3 e 5
- PORTO — R. Sá da Bandeira, 614 ☎ 38 19 30

EDITORIAL

REVISTA MENSAL
Nº 11 ANO 1
ABRIL 1989
PREÇO 350$00

PROPRIEDADE:
PUBLINFOR, Publicações e Comércio de Artigos de Informática, S.A. —

DIRECÇÃO:
Fernando Prata

COLABORADORES:
Eng. Mário Leite, Dr. Maria de Lurdes Leite, António Torres Martins, António Cardoso, Paulo Pinheiro, Miguel, João Cardoso

PRODUÇÃO GRÁFICA:
GRAFICRIA, Publicidade e Artes Gráficas, Lda. — Rua Alfredo Roque Gameiro, 21-1º Dtº Tel: 76 27 32

PUBLICIDADE E ASSINATURAS:
PUBLINFOR
Centro de Escritórios das Laranjeiras — Urbanização das Laranjeiras — Praça Nuno Rodrigues dos Santos, 7-2º Piso - Sala 13 - 1600 LISBOA Telf: 7269011 Telex 62752 Simose P Fax: 7269985

ILUSTRAÇÕES:
Franco Gomes
Miguel Angelo

FOTOGRAFIA:
Manuel Costa (Fotografia Real)

IMPRESSÃO:
Edições ASA. Divisão Gráfica) — PORTO

TIRAGEM: 11500 exemplares

PREÇO DE CAPA: 350$00

DISTRIBUIÇÃO:
ELECTROLIBER

— Nº PES. COLECT. 502009870
— Nº REG. D.G.C.S. 112959
— DEPÓSITO LEGAL Nº 20669/88

AMSTRAD PC 2386 - INTEL 80486

No momento em que a Amstrad apresenta em Portugal o PC 2386, baseado no processador INTEL 80386, este fabricante de semicondutores deixa transparecer na imprensa especializada que já começou a corrida para a aplicação do novo processador de 32 bit´s: o 80486.

Tudo parece acontecer como desde há algum tempo se prevê que aconteça: a vertiginosa evolução da alta tecnologia no domínio da informática, neste final de século, vai deixar boquiabertos todos os habitantes do planeta que se encontram mais alheios ao que acontece diariamente nesta área. Pense-se, por exemplo, que o primeiro PC IBM foi lançado no mercado apenas em 1982 e tente compreender-se, só, o que se andou desde então no caminho da velocidade e maior capacidade de armazenamento neste tipo de "engenhocas". Tudo decorreu em sete anos durante os quais cada dia foi diferente do anterior, com mais novidades, mais ideias, mais resultados de umas e de outras.

Hoje, fala-se do 80486, do NeXT, e de Steve Jobs. O SLT da Compaq começa a cair no esquecimento, e os ADD-PACK´s de Chuck Peddle (presidente da Tandon) há muito passaram à história como assunto para a imprensa especializada. Mas isto é apenas uma amostra. Nos próximos anos, não é preciso ser-se um grande génio para o afirmar, a evolução no dominio da alta tecnologia vai alterar tudo. Os utilizadores mais atentos já começaram a olhar de perto o NeXT e a imaginar o fim dos "antigos" PC´s, e... pergunte-se a propósito, será que não têm razões para isso?

Naturalmente que sim, têm já razões mais do que suficientes para tal, e num futuro muito próximo os grandes construtores vão aperceber-se de que mais uma vez os utilizadores, os analistas, e os observadores em geral estiveram certos quando deram atenção a Steve Jobs.

Um dia, daqui a alguns anos (menos do que aqueles que agora estamos a prever) a AM irá analisar aqui o novo Amstrad com unidades de leitura/escrita óptica, com impressora laser (talvez postscript), e, quem sabe, talvez concebido como laptop. Por agora resta-nos estar muito satisfeitos com o facto de podermos "saborear" a alta tecnologia em termos de computadores pessoais, através de uma receita bastante doce mas sempre standard: AMSTRAD PC 2386.

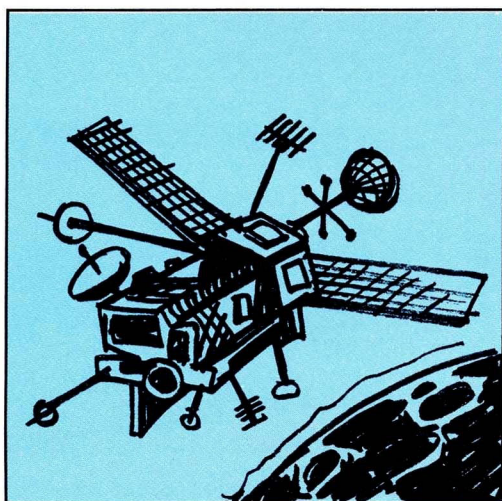
SUMÁRIO

52 DIGITALIZAÇÃO DE SONS PARA CPC

Se nós lhe dissermos que 10100101 e você nos disser que 001001010 é evidente que nós só lhe poderemos 10101001 se fizermos 101010101111. Aí sim, estaremos a digitalizar os sons com este pequeno programa para CPC.

57

CRASH! é uma adaptação para CPC 6128 do conhecidissimo jogo ´The Wall´.
Escolhemos este jogo, porque, na nossa opinião, é dos jogos mais fáceis de jogar; e como tal, um dos mais solicitados não só por quem tem o seu primeiro contacto com os computadores, mas também por todos os outros que embora possam achá-lo um jogo simples, acabam por se prender a ele ´horas e horas a fio´.

56 S.O.S. LUNA 2

Se por acaso soubesse que na galáxia 2617, precisamente junto do planeta Nixtron e na curva descrita pela cauda do cometa Killey, existe Luna 2, um pequeno satélite onde se encontram em perigo vários cientistas, o que é que pensaria fazer?
Sim, sim, exactamente tal como nós fariamos, entrava na sua nave espacial, que deixou estacionada junto ao cavalo de D. José, e procurava salvá-los.
Este jogo é um mero teste para ficar a saber se tanta iniciativa poderia obter alguns resultados.

59 CARACTERES EM TECNICOLOR

Como conseguir caracteres multicolores no ecrã, tanto em coordenadas LOCATE como em coordenadas PIXEL? Até agora tinhamos que o fazer em código-máquina, o que era pouco lógico uma vez que 99% dos programas funcionam suficientemente bem e com a rapidez necessária em BASIC.

PC 2386: UMA OPERAÇÃO "DELICATA"

A Amstrad apresentou uma nova linha de computadores pessoais. É claro que isto foi já há alguns meses. Não é novidade.
A Amstrad chegou ao nosso país com um dos modelos da nova linha de computadores. É claro que também isto foi já há alguns meses. Também não é novidade.
A Amstrad apresentou o PC 2386 no nosso país. É claro que isto foi há muito pouco tempo. Agora sim, temos novidades.

O PORQUÊ DE UMA ANÁLISE AO AT

Aproveitando o facto de poder trabalhar com um 2386 durante alguns dias, era no minímo inevitável que sobre ele fizessemos um trabalho tanto quanto possivel exaustivo, procurando trazer à superficie todos os defeitos e qualidades superiores eventualmente encontrados, mostrando ainda o que é o PC2386, numa comparação com outras máquinas idênticas mas de outros construtores, ou mesmo através de testes preparados para que os leitores possam tirar as suas próprias conclusões daquele que, tirando o lugar ao "velho" PC1640 ECD HD20, é actualmente o melhor computador pessoal com o "carimbo" Amstrad.

Durante todo o tempo em que nos sentámos frente ao teclado da máquina em análise, procurámos detectar todos os defeitos que a afectavam, ficando, no entanto, satisfeitos por todas as boas caracteristicas que nessa "luta" contra a máquina e contra o tempo se foram tornando evidentes.

ATENÇÃO! ATENÇÃO!

Depois da meia dúzia de palavras que por hábito introduzem todos os artigos, passemos então ao que interessa.

O equipamento de que dispusemos para teste era composto por uma unidade central PC 2386/65 (65 Megabytes em disco) com 4 Megabytes de RAM e uma frequência de trabalho de 20MHz, um monitor PC 12HRCD (ecrã de 12 polegadás de alta resolução e policromático), e uma drive externa de 5.25" conhecida comercialmente como uma unidade FD-10. Qualquer um dos três elementos referidos é fornecido em embalagem separada e comercializado como unidade independente, já que a configuração que utilizámos é apenas

uma das muitas possiveis com base nos mesmos elementos. Nunca é demais relembrar que, com esta nova linha de PC´s, Sugar abriu muitas portas aos utilizadores de equipamento Amstrad, ao permitir, por exemplo, que um utilizador de um PC2386 possa usar como monitor um Nec Multisync, um Philips, ou um qualquer outro monitor de um qualquer outro construtor, bem como utilizar os monitores Amstrad como componentes de uma qualquer solução informática não "Sugariana". Mas, voltemos ao trabalho.

Uma vez aberta a enorme embalagem da unidade central, as coisas começavam a tomar forma. O sentimento de surpresa inicial foi a pouco e pouco dando lugar a uma curiosidade justificada mas por vezes inoportuna que nos leva quase sempre a ligar os botões errados antes de ler a documentação certa. Neste caso fizemos, mais uma vez, precisamente isso.

Desempacotada a unidade central, extraídos os usuais "suportes" de esferovite, retirada toda a "papelada" e disquetes da caixa de cartão, eis que esta se encontrava pronta para ser armazenada para posterior transporte da máquina (tal como nos aconselham no manual - mais tarde não deixámos de ler tudo o que deviamos ter lido antes), e o CPU à espera que os restantes elementos lhe fizessem companhia.

O passo seguinte foi, como é lógico, retirar o monitor do seu confortável "ambiente de transporte". Sem aspectos interessantes esta operação foi idêntica à que antes tinhamos levado a cabo, e ainda muito semelhante à que nos esperava ao prepararmos a drive de 5.25" para iniciar os trabalhos. A diferença mais importante entre todos os "desempacotamentos" baseou-se, sem dúvida, na quantidade e qualidade da documentação encontrada, que sendo suficiente no PC 2386, poderia ser melhor no PC 12HRCD, e é deficiente na FD-10, mas sobre este aspecto falaremos mais tarde.

Com todo o equipamento em cima da mesa restava-nos começar a "ligar fios", e após duas olhadelas em volta do que tinhamos de ligar, as surpresas sucediam-se. Para além dos poucos cabos soltos (apenas 5 cabos utilizados para ligar os três componentes entre si, e alimentá-los), os poucos botões existentes, acompanhados por legendas escritas e por "bonecos", não nos deixavam quaisquer dúvidas em relação à sua finalidade, e os dip-switches eram apenas 4 (embora a finalidade destes estivesse longe de se considerar evidente). Numa apreciação geral a ligação da máquina parecia simples (e é-o na realidade), e a leitura dos "papeis". passou de imediato para segundo plano. Como podem prever isto NUNCA SE DEVE FAZER, já que por vezes os fios que parecem ter sido feitos para encaixar naquelas fichas não o foram, e os botões que parecem estar mesmo à espera que alguém os "ligue" nunca deveriam ser "ligados", resultando numa noite de São João antecipada, com algum fogo de artificio, e muito sacrificio para pagar o "estrago". Em todo o caso a nossa experiência começou assim e... felizmente começou bem.

Nas ligações não há, portanto, nada que enganar desde que se tenha um minimo de atenção a ler as legendas junto ás fichas.

Com o desejo de ver a máquina ligada tão depressa quanto o possivel, chegámos mesmo a pensar em ligar os dois cabos de alimentação do monitor e da unidade central apenas através do encosto a uma ficha eléctrica (os cabos de alimentação compostos por três condutores - positivo, neutro, e terra - não possuiem fichas, tendo apenas os terminais estanhados). Contudo, conseguimos resistir a esse impulso imprudente, e decidimos não arriscar um PC 2386 novinho, visto que um curto-circuito poderia danificar irremediavelmente o hardware. Mais alguns minutos de espera não fariam grande diferença, e as fichas lá vieram para as pontas dos cabos. A ligação à terra ficou para uma outra altura (não sentimos qualquer problema provocado por esta não-ligação do condutor de terra - amarelo com riscas verdes longitudinais).

Tudo ligado fisicamente, os dois botões essenciais são então activados. Tal como em qualquer outro PC recente e esteticamente standard (exceptuam-se os PC 1512, PC 1640, e outros do mesmo tipo), o monitor deve ser sempre ligado primeiro do que a unidade central, afim de que, quando esta for ligada, a carta gráfica possa reconhecer o tipo de monitor que vai utilizar na sessão de trabalho. Algumas cartas gráficas, tal como a EGA da Paradise fazem referência a este requisito logo nas primeiras páginas do pequeno manual que as acompanha, embora a maior parte deste tipo de dispositivos apenas deixe o facto sub-entendido. Tivemos oportunidade de constatar que a Amstrad se encontra nestes últimos casos.

Por fim, ainda nas ligações, e agora no caso especifico da FD-10, apenas devemos salientar que para além da ligação dos seus dois cabos às fichas situadas na superfície lateral esquerda da unidade central, devemos mudar o interruptor de duas posições situado na mesma superficie para a posição indicada como EXT.

FEZ-SE LUZ

Uma vez ligado, o PC 2386/65 perfaz um POST (Power On Self Test - Teste de Inicialização) visualmente idêntico ao dos anteriores modelos de PC´s do mesmo construtor, deixando aparecer no final uma imagem, também ela familiar para todos os utilizadores dos mesmos PC´s. Um ou dois segundos depois o ecrã começa a encher-se com indicações de carregamento dos vários drivers, incluidos no ficheiro CONFIG.SYS (fig. 1), e tudo termina com uma breve passagem pelo AUTOEXEC.BAT, como é hábito nestes casos.

```
@ECHO OFF
SET COMSPEC=C:\DOS\COMMAND.COM
VERIFY OFF
PATH C:\DOS;C:\WIN386
APPEND /E
APPEND C:\DOS
PROMPT $P$G
REM C:\DOS\GRAPHICS
MODE CON CP PREP=((850) C:\DOS\EGA.CPI)
KEYB UK,,C:\DOS\KEYBOARD.SYS /ID:166
MODE CON RATE=32 DELAY=1
CLS
MEM
VER
```

Figura 1a — Conteúdo do AUTOEXEC.BAT

No final de todo este tempo de espera acabamos por possuir um PC com 492480 bytes de memória de base livre, e 2 834 432 bytes de Memória Expandida (Expanded Memory) disponível (fig. 2). No fundo temos mais do que esta quantidade de memória livre, visto que os alguns drivers apenas ocuparam uma parte da RAM para a colocarem ao nosso dispor sob outras formas.

Ainda assim, as nossas tentativas para instalar o Windows 386 depois de termos arrancado a partir do disco rigido não passaram disso mesmo: tentativas (o ficheiro de instalação do windows necessita de pelo menos 512KB de RAM). A solução para instalar esta package viria a ser a utilização do comando REN para afectar o CONFIG.SYS e o AUTOEXEC.BAT, por forma a que no processo de inicialização estes ficheiros não fossem executados. Mais tarde viriamos a descobrir na página 3-22 do "User Manual and MS-DOS Guide" (um dos manuais que acompanha o computador) a indicação que a seguir traduzimos:

"IMPORTANTE: O Windows só pode ser instalado quando se utilizar no arranque a versão de MS-DOS fornecida em disquete. A versão do DOS instalada em disco rigido não deixa memória livre suficiente para o ficheiro de instalação do Windows poder funcionar."

Mais uma vez a leitura prévia da documentação facultada pela Amstrad poderia ter tido os seus frutos, se a tivessemos feito, logicamente.

A DRIVE DE 5.25"

Com o computador a funcionar em perfeitas condições, faltava-nos apenas colocar a drive de 5.25" em funcionamento. E aqui, fomos "levados" pela documentação pouco esplicita. A drive FD-10 que, em principio foi preparada para servir os utilizadores dos XT´s PC 2086, esqueceu por completo os possuidores dos AT´s. As indicações que nos são dadas nas duas ou três páginas

```
(c)1988 AMSTRAD plc
640Kb of BASE Memory
384Kb of EXTENDED Memory
64Kb of CACHE Memory


HIMEM: DOS XMS Driver, Version 2.04 - 8/17/88
Copyright 1988, Microsoft Corp.

64K High Memory Area is available.

Amstrad Memory Manager. Version 1.07 (c) Amstrad 1988
Clearing Hardware RAM pages...
Total memory pool 00C0 hardware pages.
Free: - 3F hex Handles, 0F hex Register sets, 00C0 hex Mappable pages.
Page frame at D000 hex, size 4 hex pages.
No swappable base memory.


Microsoft RAMDrive version 2.12 virtual disk D:
    Disk size: 304k
    Sector size: 512 bytes
    Allocation unit: 1 sectors
    Directory entries: 64


64K High Memory Area is available.

Amstrad Memory Manager. Version 1.07 (c) Amstrad 1988
Clearing Hardware RAM pages...
Total memory pool 00C0 hardware pages.
Free: - 3F hex Handles, 0F hex Register sets, 00C0 hex Mappable pages.
Page frame at D000 hex, size 4 hex pages.
No swappable base memory.


Microsoft RAMDrive version 2.12 virtual disk D:
    Disk size: 304k
    Sector size: 512 bytes
    Allocation unit: 1 sectors
    Directory entries: 64

Microsoft SMARTDrive Disk Cache v2.10
    Cache size: 384K in Extended Memory
    Room for 45 tracks of 17 sectors each


FASTOPEN installed

MODE prepare code page function completed
    655360 bytes total memory
    655360 bytes available
    492480 largest executable program size

   3145728 bytes total EMS memory
   2834432 bytes free EMS memory

    393216 bytes total extended memory
         0 bytes available extended memory

MS-DOS Version 4.00
C:\>
```

Figura 2 — Eco da instalação dos Drivers no ecrã

```
BREAK=ON
COUNTRY=44,,C:\DOS\COUNTRY.SYS
FCBS=20,8
FILES=20
BUFFERS=25,8
LASTDRIVE=E
SHELL=C:\DOS\COMMAND.COM /P /E:256
REM
REM Delete the REM from the following command if you will be running
REM  WIN386 at all...
REM
REM DEVICE=C:\WIN386\HIMEM.SYS
REM
REM .. and then also add a REM to the start of the following two commands !
REM
DEVICE=C:\DOS\HIMEM401.SYS
DEVICE=C:\DOS\MEMM.SYS
REM
REM But leave them alone if you will only ever run MSDOS applications,
REM  WIN (WINDOWS/286 & WINDOWS 2.03) or WIN86.
REM
DEVICE=C:\DOS\RAMDRIVE.SYS 300 /a
DEVICE=C:\DOS\SMARTDRV.SYS 576
DEVICE=C:\DOS\ANSI.SYS /X
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,1)
INSTALL=C:\DOS\KEYB.COM US,,C:\DOS\KEYBOARD.SYS
INSTALL=C:\DOS\FASTOPEN.EXE C:=(50,25)
INSTALL=C:\DOS\NLSFUNC.EXE C:\DOS\COUNTRY.SYS
```

Figura 1b — Conteúdo do CONFIG.SYS

de texto de apoio, para instalação deste dispositivo, baseiam-se na utilização de um comando **DOS** DEVICE diferente daquele que o PC 2386 possui, e aí começam as confusões quando se tenta seguir passo a passo o procedimento recomendado para a instalação da drive. Bom, resumindo, a drive FD-10 possui documentação adquada para uma instalação no PC 2086, mas muito inadequada para instalação nos modelos 286 e 386. Nestes casos, o processo de instalação é ainda mais simples (mas muito mais complicado para quem adquiriu um PC pela primeia vez, pois não é indicado em nenhum dos "papeis" que acompanham a drive), visto que basta-nos executar o ficheiro SETUP e indicar que possuimos essa drive de 5.25" com uma capacidade de armazenamento de 360KB formatados.

APLICAÇÕES E COMPLICAÇÕES

Agora sim, tudo estava perfeito, algumas aplicações em cima da mesa e as drives não paravam de rodar. Instalou-se o que se devia instalar e correu-se o que se podia correr, e os resultados foram óptimos. Das cerca de 20 aplicações que fizemos correr na máquina em causa (incluindo bases de dados, aplicações gráficas, jogos, e linguagens de programação, entre outras) apenas duas levantaram alguns problemas: o Link Master, e o teste de performance da COMPAQ.

O Link Master, package para transmissão de ficheiros via-RS232C, comercializado em especial para apoiar o Amstrad Portable Personal Computer (PPC) nesta tarefa, bloqueia depois de escolhida a primeira opção, perdendo todas as hipoteses de voltar ao seu estado normal. Aparentemente tal caracteristica deve-se ao facto de este utilitário não "gostar" do DOS 4.0 - versão de DOS utilizada pelo PC 2386/65. Se o arranque da máquina for efectuado através da drive A:, e utilizando uma disquete com uma versão anterior de DOS, este problema desaparece desde que não se utilize o disco rigido como suporte para a informação recebida (o disco está formatado em DOS 4).

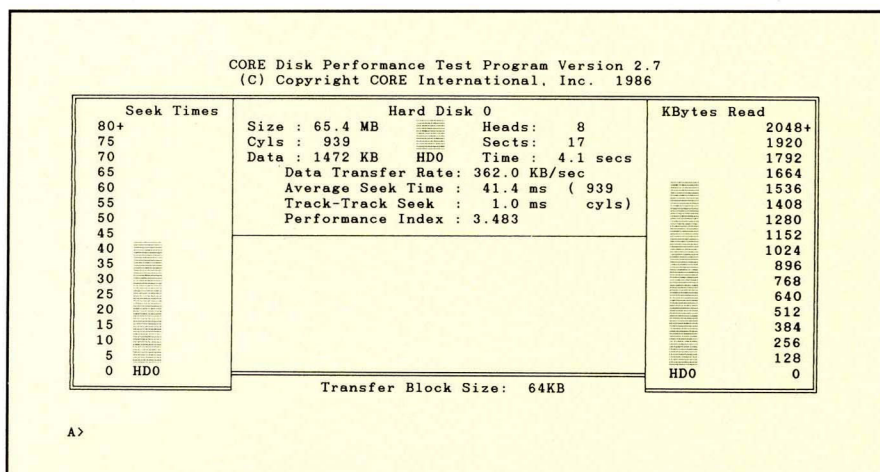
Com o teste da COMPAQ as coisas passam-se de um modo diferente. O computador "perde o teclado" depois da execução do teste "Logical Disk Access" (velocidade de acesso ao disco), continuando, no entanto, a funcionar normalmente. O grande problema advém do facto de depois desta fase o utilitário de teste da COMPAQ esperar uma entrada pelo teclado para continuar. Logo, *teste executado = computador bloqueado.*

Esquecendo estes dois "problemas", todas as outras aplicações funcionaram de um modo impecável, tornando-se tão mais interessantes quanto pesadas elas eram. Um dos casos mais espectaculares foi a inevitável BOEING GRAPHICS, que aliou a perfeição e sofisticação dos gráficos conseguidos à velocidade do 386, e mostrou o que valia no mundo da nova geração de AT´s.

VER PARAT QUERER

Sabendo, que cada vez mais as pessoas vão necessitando de tornar objectivas as suas análises para evitarem a perca de dinheiro, e poderem investir sem problemas numa tecnologia que muda todos os dias, tornando obsoletas máquinas criadas no dia anterior, não pudemos esquecer alguns testes que, embora muito faliveis, são mundialmente conhecidos como rigorosos (absurdo não é?). Assim, procurámos apresentar valores conseguidos por todos os testes minimamente conhecidos de que dispunhamos, concebendo e executando ainda alguns testes especificos em DBASE III Plus.

Com tudo o que conseguimos pretendemos acima de tudo permitir ao leitor as suas próprias conclusões sobre a nova máquina, visto que é sobretudo com base nelas que as eventuais decisões de aquisição vão, ou não, ter lugar. Afim de tornar os testes mais exactos, cada teste foi executado 25 vezes nas mesmas condições (com arranque em DOS 4.0 sem CONFIG.SYS criado pelo utilizador nem influências do AUTOEXEC.BAT), sendo em seguida testadas uma única vez duas situações de trabalho diferentes: uma de sub-aproveitamento, e outra de super-aproveitamento da máquina. A primeira destas situações foi conseguida através de um utilitário de "retardamento regulado" designado por GOSLOW, e a segunda fazendo o arranque com os CONFIG.SYS e AUTOEXEC.BAT que o computador apresentava de origem.

Os resultados demonstram bem a performance do PC em causa embora em muitos casos provem, de igual modo, o grau de falibilidade desses mesmos testes.

CORE Disk Performance Test Program (valores médios)

Os valores obtidos depois de executado o comando **GOSLOW 65535** variaram nos seguintes parâmetros:

Data : 896 KB Time : 4.0 secs

Data Transfer Rate : 222.5 KB/sec
Average Seek Time : 54.6 ms
Track-Track Seek : 3.3 ms
Performance Index : 2.331

Afectado por CONFIG.SYS e AUTOEXEC.BAT obtiveram-se os seguintes resultados nos parâmetros que sofreram alterações:

Data : 19712 KB Time : 4.0 secs

Data Transfer Rate : 4912.9 KB/sec
Average Seek Time : 41.2 ms
Track-Track Seek : 1.0 ms
Performance Index : 30.578

What Micro ? speed test program (valores absolutos)

Tempo do teste de velocidade = 74
Relação com um IBM PC = 9.39

```
A:\>sysinfo

SI-System Information, Version 3.00, (C) Copyright 1984, Peter Norton

Built-in BIOS programs dated Friday, November 18, 1988
Operating under DOS 4.00
5 logical disk drives, A: through E:
The operating system reports 640K of memory
A test of random access memory (RAM) finds:
   641K from hex paragraph 0000 to A040
    32K from hex paragraph B800 to C000
(some may be phantom memory)
BIOS signature found at hex paragraph C000
     E000
Programs are loaded at hex paragraph 27C4
following 162,880 bytes of system memory

Computing performance index relative to IBM/PC: 31.6

A:\>
```

Valores obtidos depois de executado o comando **GOSLOW 65535**

Tempo do teste de velocidade = 192
Relação com um IBM PC = 3.62

Este teste não sofreu qualquer alteração provocada pelos ficheiros CONFIG.SYS e AUTOEXEC.BAT.

THE LANDMARK CPU SPEED TEST: SPEED Version 0.99

Os resultados obtidos neste teste apontam o PC 2386 como uma máquina equivalente a um AT com uma frequência de trabalho situada entre 31.7MHz e 32.6MHz, mantendo-se19.6 a 20.2 vezes superior a um PC ou XT com uma frequência de trabalho de 4.77MHz.

O tempo que o teste demora a executar neste PC varia entre 31 e 32 milisegundos.

Depois de executado o comando **GOSLOW 65535** os valores alteraram-se de modo significativo. Assim, os valores de equivalência com o AT passaram a situar-se entre 7.2MHz e 9.3MHz, e os de comparação com o XT mudaram para 4.1 a 5.6. O teste de velocidade passou a ser executado num tempo que variava entre 111 e 152 milisegundos.

Não observámos qualquer alteração em relação aos valores normais depois de inicializarmos o sistema com o CONFIG.SYS e o AUTOEXEC.BAT já referidos.

SI - System Information, Version 3.00 (valor médio)

Computing performance index relative to IBM/PC: 30.0

Depois de **GOSLOW**

Computing performance index relative to IBM/PC: 6.6

Com o systema sob a influência dos parâmetros contidos nos CONFIG.SYS e AUTOEXEC.BAT os valores obtidos neste teste não sofrem qualquer alteração.

TESTE COMPAQ

O teste COMPAQ, é constituido por um utilitário que permite a execução (em conjunto, ou um por um) de seis testes especificos aos PC´s compatíveis:

- Rotação de gráficos
- Velocidade de acesso ao disco
- Aproveitamento do co-processador aritmético
- Procura de texto
- Velocidade de cálculo
- Velocidade de acesso à memória

Efectuado em várias máquinas, este teste é executado sempre nas mesmas condições, sendo desejável, para que os dados sejam reais, que o computador arranque com a própria disquete de teste. Esta disquete, formatada em COMPAQ Personal Computer DOS version 3.31, foi a única excepção em termos de sistema operativo suporte dos testes efectuados. Todos os testes anteriores foram efectuados sobre DOS 4.0.

DBASE III PLUS: OS TESTES E OS TEMPOS

Depois de executarmos todos os testes já referidos achámos que tinha algum interesse poder comparar a performance da nova máquina Amstrad com a da sua "antiga" maior máquina através da execução de pequenas tarefas. O DBASE impunha-se aqui, por várias razões, inclusivé porque possuíamos uma base de dados de trabalho com 3528 registos, o que evitava estarmos a criar "ambientes de trabalho falsos". Decidimos trabalhar com aquilo a que estavamos habituados e os resultados foram, e são, uma excelente amostra das capacidades do 80386 para todos os leitores que não estão familiarizados com o poder e/ou caracteristicas deste processador.

Teste 1

```
* PC 2386/65     TEMPO=00.39.76
* PC 1640 HD20   TEMPO=04.49.63

wait 'Prima uma tecla para
iniciar o teste ' to MKEY
use ce
index on cedesigemp to cedemp
use
? chr(7)+'Teste concluido.'
```

Teste 2

```
* PC 2386/65     TEMPO=04.10.40
* PC 1640 HD20   TEMPO=+2 horas
wait 'Prima uma tecla para
iniciar o teste ' to MKEY
use ce
do whil recn()<10000
   appe blan
endd
use
? chr(7)+'Teste concluido.'
```

Teste 3

```
* PC 2386/65     TEMPO=01.16.18
* PC 1640 HD20   TEMPO=10.54.15
wait 'Prima uma tecla para
iniciar o teste ' to MKEY
use ce
dele all for recn()>3528
pack
use
? chr(7)+'Teste concluido.'
```

Teste 4

```
* PC 2386/65     TEMPO=00.20.92
                 RECORDS=273
* PC 1640 HD20   TEMPO=02.06.23
wait 'Prima uma tecla para
iniciar o teste ' to MKEY
use ce
list all for 'Z' $ CEDESIGEMP
use
? chr(7)+'Teste concluido.'
```

Teste 5

```
* PC 2386/65    TEMPO=03.13.88
* PC 1640 HD20  TEMPO=16.58.57

wait 'Prima uma tecla para
iniciar o teste ' to MKEY
use ce
list
use
? chr(7)+'Teste concluido.'
```

```
C:\>ps

Allocated Memory Map - by TurboPower Software - Version 2.1

 PSP  blks bytes owner    command line        hooked vectors
----- ---- ----- -------- ------------------- ---------------------------
0008   1   76880 config
1E77   1    6128 N/A      US,,C:\DOS\KEYBO... 09
1FF7   1   10096 N/A      C:=(50,25)
226F   1    2752 N/A      COU
231C   1    5344 N/A      /NC
246B   2    5952 N/A                          22 24 2E
25E1   1    7712 N/A      /E                  21 2F
27C4   2  492480 free

block   bytes   (Expanded Memory)
-----  ------
    0  311296
    1       0
 free 2834432
total 3145728

C:\>


C>ps

Allocated Memory Map - by TurboPower Software - Version 2.1

 PSP  blks bytes owner    command line        hooked vectors
----- ---- ----- -------- ------------------- ---------------------------
0008   1   12256 config
0EA3   2    5856 command                      22 24 2E
1017   2  589456 free

C>
```

Espaço de memória ocupado pelos diversos drivers e programas/comandos residentes

HARDWARE

Sem resistir à tentação abrimos o PC 2386.

O processo de abertura da caixa exterior é relativamente simples. Começamos por retirar a tampa lateral dos slots, fazendo-a deslizar da frente para a traseira do computador. Seguidamente, desapertamos o parafuso que se encontra a meio da tampa superior dos slots, fazendo deslizar esta tampa no mesmo sentido da que antes retirámos. O passo seguinte consiste então em retirar as duas "rolhas" de plástico que escondem os parafusos situados nos cantos superiores frontais da unidade central, e desapertar esses mesmos parafusos, bem como todos os outros que se encontram visiveis neste momento (não devemos esquecer os parafusos que seguram as "chapinhas" dos slots).

Feito isto, a maior parte do "esqueleto" do computador fica exposta aos olhos dos curiosos. Podemos nesse momento começar a dividir o espaço fisico dentro do computador em três partes distintas: os slots, a fonte de alimentação, e as unidades de suporte magnético de informação.

Na primeira área por nós referida podemos encontrar 5 slots dispostos no sentido paralelo à frente do computador, dos quais 4 são slots full-size e 1 é slot tipo XT. Este último slot tem, contudo, todos os sinais no local certo da placa para poder ser um slot full-size, o que nos levou a ficar com algumas dúvidas sobre as razões que levaram Sugar a cortar o slot ao meio. Assim, se pretender mais um slot full-size no seu PC2386 basta-lhe retirar a ficha DIL de 62 contactos e substitui-la por uma outra de 98. Tudo muito simples, desde que o construtor não possua razões válidas para evitar este tipo de slot na placa.

A fonte de alimentação, que constitui a divisão central do espaço interior, não tem muito de interessante. É uma fonte de alimentação, esteticamente como muitas outras, possuindo uma ventoinha na sua face lateral direita, e o interruptor de pressão que liga a unidade na face esquerda. Com uma caixa metálica que envolve por completo a fonte servindo-lhe de blindagem, os segredos sobre esta parte da máquina ficaram quase todos por revelar.

A última divisão referida (ou a primeira, dependendo de onde se situa o observador), é a que é formada pelas unidades de armazenamento de informação, ou, de uma forma mais vulgar, pelas drives. A unidade que "descascámos" ficava com este espaço completamente preenchido, sem que uma eventual terceira drive interna pudesse ser adaptada (de origem este computador é fornecido com uma drive de 3.5" e um disco rigido de 65MB).

Para além dos elementos referidos ficámos ainda bastante surpreendidos ao constatar que o PC 2386 baseia a sua carta gráfica num circuito integrado da Paradise Systems (uma subsidiária da Western Digital), o que faz com que esta carta gráfica seja muito semelhante à carta VGA dos PC´s Philips, por exemplo, e justifica todas as possibilidades de inicialização do sistema num determinado modo gráfico previamente definido (ex. **DEVICE DISPLAY HERC1/B**).

Voltando um pouco atrás, precisamente ao ponto onde falámos das drives, não dissemos ainda que também não resistimos a extrair o disco rigido da blindagem que o escondia dos olhares curiosos, e que depois de alguns segundos de trabalho ficámos a saber que este PC 2386/65 possuia um disco de 65MB da Seagate com a referência ST-277R MLC0. Algum trabalho de pesquisa e eis as caracteristicas do disco em causa:

Capacidade de armazenamento formatada (em Megabytes) : 65.5
Tempo de acesso "track-to-track" : 8 milisegundos
Tempo médio de acesso : 40 milisegundos
MTBF : 35000
Limites de temperatura ambiente : 15º a 45ºC
Consumo : 11 Watts
Formato : 5.25 "meia altura".

Visto que estavamos com "as mãos na massa" aproveitámos e desmontámos também a drive de 3.5".

Em todos os aspectos igual ás drives dos PPC´s, esta pequena drive de 3.5" dava pelo nome de 0SDA-45A e apresentava-se deste modo como um modelo de drive Citizen, beneficiando como quase todos os componentes recentes deste tipo, de uma construção com base em tecnologia de montagem de superfície. Este tipo de montagem de componentes, utilizada pela Amstrad em vários outros circuitos dentro da mesma unidade central, apresenta aliás uma particularidade interessante: nos locais onde o espaço era algo a ter em consideração durante a montagem, os circuitos integrados não possuem os pinos "espalmados" contra o circuito impresso, tendo-se optado antes por

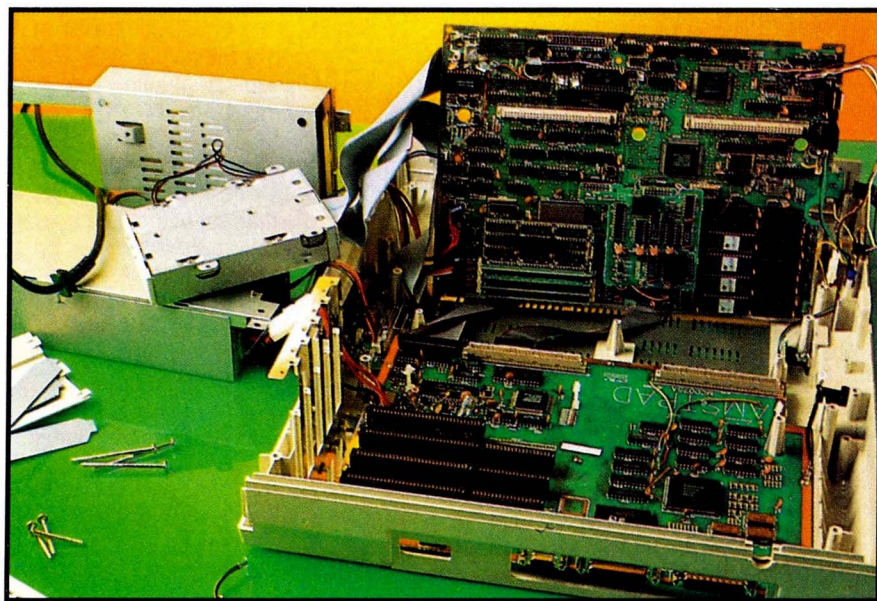
voltá-los para dentro e por debaixo do próprio ciruito integrado, poupando deste modo alguns milimetros no espaço da placa. Esta forma de dobrar os pinos dos integrados foi para nós uma novidade agradável.

Por fim, neste "capitulo" do hardware falta apenas referir a abertura existente debaixo da unidade em causa. Com efeito este 386 possui uma placa de metal na base, que pode ser retirada mediante o desaperto de apenas três parafusos, afim de permitir a fácil inserção do processador aritmético 80387, e possibilitar a alteração de algumas pontes para configuração permanente por hardware (veja-se as páginas 3-26 a 3-29 do "User Manual and MS-DOS Guide").

O MONITOR

O monitor utilizado durante os testes foi como já referimos um Amstrad PC 12HRCD que proporcionou sempre uma imagem perfeita em todos os modos de trabalho, sem "flickering" nem bordos destorcidos. Agora num estilo muito mais "IBMico", o monitor Amstrad analisado mantêm, no entanto, várias semelhanças com os modelos da série 1000 do mesmo construtor. No manual do utilizador fornecido com a unidade central os diversos desenhos em que o monitor nos aparece fogem mesmo muito à realidade, ao apresentarem caracteristicas que não sendo comuns a esta nova linha de monitores, se podem constatar nos antigos modelos PC´s 1512 e 1640.

Sob o aspecto estético, estes monitores poderiam ter sido mais bem "pensados". A deslocação dos poucos botões de controlo para o painel frontal, ou mesmo o acrescento de mais alguns potenciometros ou comutadores para conseguir um maior controlo do monitor, por exemplo, seriam dois dos factores que talvez tivessemos em atenção ao desenhar esta peça. Como é lógico, nestes casos o custo de qualquer alteração pesa sempre nas decisões e talvez Sugar tenha começado a contar os tostões (se bem que nos custe bastante a acreditar nisso).

Voltando a falar da qualidade de imagem obtida neste monitor, podemos relembrar os mais entendidos nestes assuntos que o PC 12HRCD possui um *tridot pitch* de 0.28 milimetros, o que o equipara, por exemplo, ao Nanao Flexscan, ou ao Princeton Graphic Systems Ultrasync, sendo batido neste parâmetro por muito poucos monitores disponiveis no mercado - apenas nos lembramos do Sony Multiscan (0.26 mm). Relembramos ainda os leitores que quanto menor for o *tridot pitch* de um monitor maior qualidade terá a imagem por ele visualizada.

SOFTWARE E DOCUMENTAÇÃO

O complemento de software que acompanha o PC 2386, constitui mais um dos pontos positivos encontrados. Largando definitivamente os ambientes de trabalho da Digital Research (que poderiam continuar a subsistir nos 2086), a Amstrad decidiu voltar-se para a Microsoft e permitir que os utilizadores das suas máquinas possam disfrutar o poder de um Windows 386, por exemplo. Consequência desta mudança, o PC 2386 surge-nos no mercado acompanhado por 9 disquetes de 3.5" (ao contrário das 8 que são referidas no manual do utilizador). As referidas disquetes estão distribuidas entre o Windows 386 (6 disquetes), o MS-DOS 4.0 (2 disquetes), e uma disquete suplementar para optimização do sistema e solução de situações especificas. Esquecendo, por agora, os dois primeiros conjuntos de disquetes, que por si só dariam assunto para um artigo tão estenso como este, apenas vamos sublinhar a importância da disquete suplementar, que para além de permitir a optimização do sistema em diversas situações e ambientes de trabalho (rede AMSNOS, etc.), possibilita ainda a inicialização do sistema a partir da drive B: (drive externa de 3.5" ou 5.25") afim de assegurar que o utilizador consegue "correr" software protegido em 5.25".

Antes de falarmos da documentação, não queremos deixar de sublinhar o nosso contentamento com a alteração no driver do rato, que tantos problemas tinha causado aos utilizadores dos anteriores PC´s da série 1000, de modo a permitir que o relógio interno do PC se mantenha exacto independentemente da aplicação que este esteja a executar.

Por fim, uma breve nota sobre a documentação.

Composta por 5 manuais, a documentação de apoio da máquina em análise encontra-se bem estruturada, e permite a fácil detecção de informações, e conselhos, sobre a maior parte dos assuntos que um utilizador necessita conhecer para trabalhar com o hardware que adquiriu. O único alerta vai para as gralhas de composição, que são em demasia, e o único "é pena" vai para o facto de toda a documentação se encontrar em inglês em lugar do nosso "costumeiro dialecto", o que... é pena.

CONCLUSÃO

Sem atingir um nível de performance superior, ou mesmo idêntico, aos modelos de high-end da Compaq, o primeiro 80386 da Amstrad vai com certeza fazer sucesso no nosso mercado, não só porque nem todos necessitam de possuir um computador que se mantenha no topo da pirâmide como o computador mais "possante" dentro da sua classe, mas também porque a maior parte dos utilizadores não podem, ou não querem, suportar os custos que tal posse poderia implicar. Qualquer modelo da linha IBM PS/2 não "chega aos calcanhares" do menino bonito da Compaq (se é que ele tem calcanhares), e no entanto a IBM não parece estar muito aborrecida com isso, afirmando a importância das suas máquinas através do bus MCA, por exemplo.

O PC 2386 vai constituir uma solução informática com base no mais potente processador da INTEL, mas simultâneamente vai constituir uma solução informática com base no preço Amstrad, que continua a ser um dos mais baixos do mercado. Esta mistura dos dois ingredientes fundamentais para a posição que a Amstrad actualmente ocupa no mercado europeu, vai garantir, pelo menos, que Sugar não passe fome nos próximos meses, e que um maior número de utilizadores possa saciar a sua fome da alta tecnologia.

Fernando Prata

COMPAQ Performance Measurement Demo

Version 11.63

Machine	Calc. Speed	Memory Access	Disk Access	Graphics Rotation	Text Search	Coproc Speed
COMPAQ DESKPRO 386/25 Mod 300	1.78	2.22	3.79	5.30	3.23	2.34
COMPAQ DESKPRO 386/25 Mod 110	1.78	2.22	5.10	5.30	3.23	2.34
COMPAQ DESKPRO 386/20 Mod 130	2.20	2.78	4.17	6.01	3.82	2.91
COMPAQ DESKPRO 386/20e Mod 110	2.20	2.78	5.18	5.32	3.61	2.96
COMPAQ DESKPRO 386/20e Mod 40	2.20	2.78	7.09	5.32	3.61	2.96
COMPAQ PORTABLE 386 Model 100	2.64	3.15	6.62	4.54	6.38	3.17
COMPAQ PORTABLE 386 Model 40	2.64	3.15	8.14	4.54	6.37	3.17
COMPAQ DESKPRO 386 Model 40	3.27	3.98	7.11	8.03	5.50	3.97
COMPAQ DESKPRO 386s Model 40	3.94	4.57	7.02	8.51	6.13	4.33
COMPAQ DESKPRO 386s Model 20	3.94	4.58	7.65	8.51	6.13	4.33
COMPAQ PORTABLE III Model 40	5.87	6.02	7.94	7.65	9.63	13.04
COMPAQ DESKPRO 286 Model 40	5.87	6.02	7.45	10.45	7.71	13.04
COMPAQ SLT/286 Model 40	5.88	6.04	6.92	9.65	7.20	9.44
COMPAQ SLT/286 Model 20	5.88	6.04	7.61	9.65	7.20	9.44
IBM PS/2 Model 80-311	2.62	3.31	6.52	7.92	7.04	3.45
IBM PS/2 Model 80-111	2.62	3.29	6.80	7.90	7.03	3.45
IBM PS/2 Model 70-121	2.63	3.32	6.48	7.91	8.15	3.52
IBM PS/2 Model 80-071	3.43	4.06	7.52	9.17	7.96	4.34
IBM PS/2 Model 70-E61	3.33	4.17	8.01	9.28	9.27	4.41
IBM PS/2 Model 60-041	6.94	7.15	8.44	13.01	11.14	10.49
IBM PS/2 Model 50Z-031	5.34	5.73	9.33	11.41	11.94	9.98
IBM PS/2 Model 50-021	6.98	7.20	15.51	13.10	11.26	10.53
IBM PS/2 Model 30 286	6.92	7.09	18.85	13.32	13.65	14.07
COMPAQ DESKPRO 386/25 Mod 300	1.78	2.22	.80	5.30	3.23	2.34
COMPAQ DESKPRO 386/25 Mod 110	1.78	2.22	.74	5.30	3.23	2.34
COMPAQ DESKPRO 386/20 Mod 130	2.20	2.78	.63	6.01	3.82	2.91
COMPAQ DESKPRO 386/20e Mod 110	2.20	2.78	.74	5.32	3.61	2.96
COMPAQ DESKPRO 386/20e Mod 40	2.20	2.78	1.03	5.32	3.61	2.96
COMPAQ PORTABLE 386 Model 100	2.64	3.15	.83	4.54	6.38	3.17
COMPAQ PORTABLE 386 Model 40	2.64	3.15	1.30	4.54	6.37	3.17
COMPAQ DESKPRO 386 Model 40	3.27	3.98	1.26	8.03	5.50	3.97
COMPAQ DESKPRO 386s Model 40	3.94	4.57	1.10	8.51	6.13	4.33
COMPAQ DESKPRO 386s Model 20	3.94	4.58	1.23	8.51	6.13	4.33
COMPAQ PORTABLE III Model 40	5.87	6.02	1.42	7.65	9.63	13.04
COMPAQ DESKPRO 286 Model 40	5.87	6.02	1.31	10.45	7.71	13.04
COMPAQ SLT/286 Model 40	5.88	6.04	.96	9.65	7.20	9.44
COMPAQ SLT/286 Model 20	5.88	6.04	1.28	9.65	7.20	9.44
IBM PS/2 Model 80-311	2.62	3.31	.77	7.92	7.04	3.45
IBM PS/2 Model 80-111	2.62	3.29	.83	7.90	7.03	3.45
IBM PS/2 Model 70-121	2.63	3.32	.79	7.91	8.15	3.52
IBM PS/2 Model 80-071	3.43	4.06	.87	9.17	7.96	4.34
IBM PS/2 Model 70-E61	3.33	4.17	.94	9.28	9.27	4.41
IBM PS/2 Model 60-041	6.94	7.15	1.15	13.01	11.14	10.49
IBM PS/2 Model 50Z-031	5.34	5.73	1.09	11.41	11.94	9.98
IBM PS/2 Model 50-021	6.98	7.20	1.85	13.10	11.26	10.53
IBM PS/2 Model 30 286	6.92	7.09	2.01	13.32	13.65	14.07
PHLIPS NMS9115 C/ CARTA EGA	21.71	21.19	17.81	39.61	46.51	41.41
PHILIPS TC100	17.35	16.94		26.01	30.17	33.18
PHILIPS NMS 9125 AT 286	8.39	9.38	15.12	14.15	17.04	17.36
AMSTRAD PC2086	16.70	16.15	24.28	38.19	30.31	62.94
PPC 512 C/ MONITOR ECD	12.78	15.13		18.72	25.81	34.83
Amstrad PC2386/65	2.30	2.91	9.11	6.16	5.73	46.83
COM CONFIG.SYS + AUTOEXEC.BAT	**2.30**	**2.91**	**1.21**	**6.16**	**5.73**	**46.83**

Reprodução de gravura a água-forte de José Faria

TACTO

O tacto. A capacidade do Homem agarrar o sentido das coisas. De fazer contacto entre o imaginário e o real.

A Sopsi entende que a informática está de maõs dadas com a realidade sensível. Assim, procurou cuidadosamente os seus contornos essenciais. Apalpou o terreno. Sentiu as melhores oportunidades do mercado. E agiu.

Os 10 000 contos iniciais do capital social da Sopsi passaram entretanto para 200 000 contos e, mais recentemente, para 350 000 contos. As vendas que, em 1983, foram de 45 000 contos, atingiram os 2 400 000 contos, em 1987, estimando-se que possam ultrapassar a barreira dos 3 000 000 contos, já este ano.

O resultado líquido deverá crescer de 80 000 contos, em 1987, para cerca de 190 000 contos, no ano em curso.

E, nos últimos dois anos, subiu da 19.ª posição para o 5.º lugar das maiores empresas de informática no País, ocupando já a 2.ª posição entre as que possuem capital inteiramente nacional.

Por outro lado, representando e comercializando algumas das melhores marcas do mercado, e sendo maioritária ou detendo significativas participações financeiras em importantes empresas do sector, a Sopsi afirma-se como a empresa que Hoje mexe com o Futuro.

O Sentido da Informática

Sopsi-SOCIEDADE PORTUGUESA DE SISTEMAS DE INFORMÁTICA, S. A.
AV. DA BOAVISTA, 2881-2.º — 4100 PORTO

mIssionários, cAnibais e PROgramação LOGica

PORQUÊ O PROLOG

Diz-se hoje em dia que o Prolog é a linguagem da inteligência artificial (IA). Será que o Prolog tem algum mecanismo mais ou menos mágico sem o qual se torna impossível fazer uma aplicação de inteligência artificial? A resposta é obviamente não. Não há nada que se faça em Prolog que não se faça também em Pascal, C, Fortran ou qualquer outra linguagem. A questão é apenas a da adequação da ferramenta ao problema a resolver e a verdade é que o Prolog é uma óptima ferramenta no domínio da manipulação de objectos e de relações entre eles. Isto torna-o numa linguagem de eleição na programação de inteligência artificial e não numérica em geral. Note-se ainda que o Prolog é muito mais novo que a disciplina de IA e que a linguagem com maior nº de linhas de código escritas neste domínio é sem dúvida o Lisp.

UMA LINGUAGEM DECLARATIVA

Enquanto que as linguagens convencionais têm um caracter imperativo e procedimental, o Prolog é essencialmente declarativo. Ao contrário de um programa Pascal que dita uma determinada ordem de acções a desempenhar, um programa Prolog limita-se idealmente a exprimir um conjunto de factos e regras no domínio do problema em causa. Isto torna-o numa linguagem mais próxima de nós e por isso de mais alto nível.

Será difícil arranjar um exemplo de uma linguagem de programação que dê mais gozo aprender que o Prolog. Aprender Prolog não exige decorar um largo número de regras sintácticas e por outro lado, obriga a uma maneira de pensar diferente dentro do contexto das linguagens de programação. Isto torna-o aliciante, tanto para as pessoas que já dominam outras linguagens, como para aquelas que se pretendem agora iniciar neste campo.

Quanto aos mecanismos do Prolog, eles são essencialmente três: a unificação, estruturas de dados baseadas em árvores e o backtracking. Apenas estes três mecanismos tornam-no contudo numa linguagem extremamente poderosa.

A NOSSA PRIMEIRA BASE DE DADOS

Imaginemos que queríamos exprimir algumas relações de parentesco entre pais e filhos. O que se faz é escrever num ficheiro um conjunto de factos que as representem, por exemplo:

— Em comentário estão as relações expressas.

```
pai( joao, luis ).— o pai do João é o Luis.
pai( manuel, francisco )— o pai do Manuel é o
Francisco
pai( pedro, francisco )— o pai do Pedro é o
Francisco
pai( rui, joao )— o pai do Rui é o João
```

O passo seguinte será invocar o interpretador Prolog e consultar este ficheiro. Após fazermos isto podemos então interrogar o Prolog acerca dos factos expressos:

```
? - pai( joao, luis )
yes

? - pai( joao, rui )
no

? - pai( pedro, X )
 X = francisco ->.
yes
```

Note-se que em Prolog uma variável começa sempre por letra maiúscula.

O mecanismo utilizado pelo Prolog nesta situação é o da unificação. Quando se lança o goal (para utilizar a terminologia do Prolog) *pai(pedro, X)*, a variável X não tem valor algum e diz-se que está não instanciada, estado em que unifica com qualquer coisa.

Estamos então agora em condições de perceber o que se passou: O Prolog percorreu a base de dados tentando unificar *pai(pedro, X)* com um dos factos desta. 'As duas primeiras tentativas falhou pois pedro não faz o chamado matching nem com joao nem com manuel. Mas à terceira sucede e unifica X com francisco.

É evidente que também podemos perguntar de quem é que o francisco é pai:

```
? - pai( X, francisco ).
X = manuel ->
```

com um ; o Prolog faz o chamado backtracking e tenta uma nova solução:

```
X = manuel ->;
X = pedro ->;
no
```

UMA REGRA PROLOG

Então e agora se quisessemos alargar a nossa base de dados para relações entre avôs e netos ? Uma hipótese seria acrescentar a esta, novos factos que exprimissem esse parentesco. No nosso caso particular:

```
avo( rui, luis ).
```

Mas agora de cada vez que inseríssemos novos *pai*'s teríamos que procurar novas relações *avo* ou a base de dados ficaria incoerente. Por aqui não vamos lá ... O que dava jeito era definir a relação *avo* à custa da relação *pai* e encarregar o Prolog de encontrar os avôs na base de dados. É muito fácil fazer isto em lógica:

```
pai( X, Y ) e pai( Y, Z ) implica avô( X, Z )

a tradução para Prolog é imediata:

avo( X, Z ) :- pai( X, Y ), pai( Y, Z ).
```

e agora o Prolog já será capaz de responder a:

```
? - avo( X, luis ).
X = rui ->.
yes
```

E pronto escrevemos a nossa primeira regra Prolog. Podemos agora perceber porque razão se diz que o Prolog é uma linguagem de programação lógica. Cada uma das regras constitui afinal um pequeno axioma que o Prolog utiliza como demonstrador de teoremas que acaba por ser. Segundo este prisma, os goals não passarão então de teoremas que o Prolog se encarrega de demonstrar para nós.

LISTAS, LISTAS E LISTAS DE LISTAS

Uma estrutura que se usa intensamente em manipulação simbólica é a lista. Uma lista tem um número arbitrário de elementos arbitrários. Usando a notação Prolog podemos ter as seguintes listas:

```
[ joao, manuel, pedro ] — Uma lista de pessoas

[ joao, [ pedro, manuel ], [ manuel, [] ] ] —
uma lista de listas

[ joao, X, conceicao ] — Uma lista em que um
dos elementos é uma variável

[ H1, H2 | T ] — Esta notação é um pouco
diferente e representa uma lista em que o
primeiro elemento é H1, o segundo é H2 e o resto
da lista é T
```

Como se vê um elemento de uma lista pode inclusivamente ser uma lista e assim sucessivamente, concluindo-se pois que a lista é uma estrutura de dados recursiva. A recursividade é de resto um mecanismo de programação fundamental no Prolog.

Vejamos então como escrever um pequeno programa Prolog que faça a concatenação de duas listas:

```
append( [], X, X ).
```

Esta primeira regra diz simplesmente que se juntarmos à lista vazia uma lista X, obtemos a lista X. Falta-nos dizer como juntar duas listas em que a primeira é não vazia:

```
append( [ H | T ], X , [ H | T2 ] ) :-
 append( T, X, T2 ).
```

Esta segunda regra diz o seguinte:

Se quando a T juntamos X obtemos T2, então quando a *[H | T]* juntamos X havemos de obter *[H | T2]*.

E pronto, *append* está pronto a juntar qualquer par de listas:

```
? - append( [ a, b ], [ b, d ], Y ).
Y = [ a, b, b, d ] ->.
yes
```

No entanto se olharmos mais atentamente para o programa, vemos que *append(X, Y, Z)* não se limita a definir Z em função de X e Y. O que nós fizemos afinal foi exprimir uma relação entre três listas e gostaríamos pois que o Prolog funcionasse de acordo com isso. Ora o que torna o Prolog irresistivelmente atraente é o facto de manter o seu caracter declarativo nestas situações, acabando por nos surpreender ao resolver goals como:

```
? - append( X, [ b, d ], [ a, b, b, d ] ).
X = [ a, b ] ->.
yes

? - append( [ a ], X, [ a, b, b, d ] ).
X = [ b, b, d ] ->.
yes

? - append( [ a, b ], [ b, d ], [ a, b, b, d
] ).
yes

? - append( X, Y, [ a, b, b, d ] ).
X = []
Y = [ a, b, b, d ] ->;

X = [ a ]
```

```
Y = [ b, b, d ] ->;

X = [ a, b ]
Y = [ b, d ] ->;

X = [ a, b, b ]
Y = [ d ] ->

X = [ a, b, b, d ]
Y = [] ->;
no
```

O processo pelo qual o Prolog resolve estas situações não tem nada de transcendente mas sai um pouco fora do âmbito deste artigo. Numa próxima edição da AM veremos como isso é feito.

DE VOLTA À INTELIGÊNCIA ARTIFICIAL

Falámos já de inteligência artificial e de aplicações de inteligência artificial. Não vamos contudo agora aqui discutir se os computadores podem ou não actuar com inteligência. Adoptaremos pelo contrário a atitude de Pedro Guerreiro que no 1º capítulo de 'Pascal Técnicas de Programação' acaba por concluir que ' saber se os computadores pensam ou não, é uma questão linguística, não informática ', isto após citar a igualmente deliciosa frase de Dijkstra: ' saber se os computadores pensam é tão importante como saber se os submarinos nadam'. Tentaremos contudo dar uma ideia (os ingleses diriam flavour) do que é um problema na área de IA através da resolução de uma charada, o problema dos missionários e canibais.

3 MISSIONÁRIOS + 3 CANIBAIS = PROLOG

Três missionários e três canibais encontram-se na margem esquerda de um rio e pretendem atravessá-lo para o outro lado. Para tal dispõem de um pequeno barco com lotação máxima para dois indivíduos. Acontece que após já vários dias de jornada, os canibais ainda se mantêm fieis aos seus bons velhos costumes. Consequentemente para que os missionários não acabem por atravessar o rio na barriga dos canibais têm que se manter sempre, senão em vantagem, pelo menos em igualdade numérica em qualquer das margens. Por exemplo se a dado passo tivermos um missionário e um canibal do lado esquerdo e dois missionários e dois canibais do lado direito, não podemos mandar um dos canibais à margem esquerda com o barco, pois isso teria consequências menos agradáveis para o missionário. Por outro lado, o barco nunca pode atravessar sózinho senão vai com a corrente e o problema perde a graça. Como atravessar então todos eles sãos e salvos ?

Este é um problema típico de pesquisa onde em cada estado temos um conjunto de estados sucessores possíveis. Representemos então qualquer estado por uma estrutura *estado(Lado, Missionarios, Canibais)* onde Lado diz o lado onde o barco está, Missionários o número de missionários desse lado e Canibais o número de canibais também desse lado. Desta forma o estado inicial será:

```
estado( esq, 3, 3 ).
```

e o final:

```
estado( dir, 3, 3 ).
```

Alguns sucessores do estado inicial serão:

```
estado( dir, 1, 1 ) — Passou um missionário e
um canibal

estado( dir, 0, 2 ).— Passaram dois canibais
```

o sucessor *estado(dir, 2, 0)* não é válido pois significa que do lado esquerdo ficaram três canibais e apenas um missionário. A solução do problema é então arranjar uma sequência válida de viagens que nos leve do estado inicial ao final. Uma vez que cada estado tem sempre um conjunto de sucessores possíveis , acabamos por obter uma árvore por onde vamos navegar à procura do estado final.

UMA SOLUÇÃO

Está-se mesmo a ver que vamos precisar de uma regra que nos permita decidir sobre a validade de um estado. O melhor é pois escrevê-la já:

```
bomLado( Canibais, Missionarios ) :-
 Canibais =< Missionarios.
bomLado( Canibais, Missionarios ) :-
 Missionarios = 0.
```

Uma outra coisa que vai ser com certeza necessária é achar o sucessor de um estado. Para tal vamos definir o conjunto de operadores que se podem aplicar a um estado e que serão neste caso , a descrição dos indivíduos que vão no barco em cada viagem:

```
oper( passa( 2, 0 ) ).— Dois canibais
oper( passa( 1, 0 ) ).— Um canibal
oper( passa( 1, 1 ) ).— Um canibal e um
missionário
oper( passa( 0, 1 ) ).— Um missionário
oper( passa( 0, 2 ) ).— Dois missionários
```

O próximo passo será escrever uma regra que aceite o estado inicial e devolva a lista das viagens. Como sempre vamos limitar-nos a traduzir o enunciado do problema e encarregar o Prolog de fazer o resto:

```
resolver( Estado, [] ) :- solucao(
Estado ).
resolver(        Estado,        [
Operador|Caminho ] ) :-
 oper( Operador ),
   possivel(  Operador,   Estado,
EstadoNovo ),
 resolver( EstadoNovo, Caminho ).
```

possivel é o predicado que verifica se um determinado operador se pode aplicar a Estado e que em caso afirmativo devolve o estado resultante, EstadoNovo.

A primeira regra diz-nos simplesmente que resolver um estado que já é solução é não fazer nada. A segunda diz-nos o seguinte:

[Operador | Caminho] é uma sequência de viagens que leva de Estado à solução se EstadoNovo for um sucessor possível de Estado e Caminho uma sequência que leva de EstadoNovo à solução.

Simples não é ? O problema é que não funciona. Tal como está nada impede o Prolog de entrar em ciclo, ou seja, repetir indefinidamente as mesmas sequências de viagens. Mais específicamente, o que aconteceria é que o Prolog escolhia sempre o primeiro operador e passava a vida a passear dois canibais de uma margem para a outra.

UMA SOLUÇÃO QUE FUNCIONA

Para evitar que isto aconteça vamos andar sempre com uma lista atrás que nos diz os estados por onde já passámos. Sempre que chegamos a um estado novo verificamos se ele já se encontra na lista, e em caso afirmativo abandonamos imediatamente essa hipótese de pesquisa. É fácil ver se um elemento pertence a uma lista:

```
pertence( X, [ X | T ] ).
pertence( X, [ Y | T ] ) :-
 pertence( X, T ).
```

ou seja, um elemento para pertencer a uma lista ou é o primeiro elemento dessa lista ou pertence à cauda desta.

A versão melhorada de *resolver* será então:

```
resolver( Estado, [], _ ) :- solucao( Estado
).
resolver( Estado, [ Operador|Caminho ], Memo
) :-
 oper( Operador ),
 possivel( Operador, Estado, EstadoNovo ),
 not pertence( EstadoNovo, Memo ),
 resolver( EstadoNovo, Caminho, [ EstadoNovo
| Memo ] ).
```

LISTAGEM COMPLETA DO PROGRAMA DOS MISSIONARIOS E CANIBAIS

```
resolvo( Estado, Caminho ) :-
  resolver( Estado, Caminho, [ Estado ] ).

resolver( Estado, [], _ ) :- solucao( Estado ).
resolver( Estado, [ Operador|Caminho ], Memo ) :-

  oper( Operador ),
  possivel( Operador, Estado, EstadoNovo ),
  not pertence( EstadoNovo, Memo ),
  resolver( EstadoNovo, Caminho, [ EstadoNovo | Memo ]
).

possivel( passa( Bcan, Bmis ), estado( Lado, Can, Mis ),
          estado( LadoF, CanF, MisF )  ) :-
  Can >= Bcan,
  Mis >= Bmis,
  CanF is 3 - Can + Bcan,
  MisF is 3 - Mis + Bmis,
  bomLado( CanF, MisF ),
  DirCanF is 3 - CanF, DirMisF is 3 - MisF,
  bomLado( DirCanF, DirMisF ),
  trocaLado( Lado, LadoF ).

bomLado( Canibais, Missionarios ) :-
  Canibais =< Missionarios.
bomLado( Canibais, Missionarios ) :-
  Missionarios = 0.

trocaLado( esq, dir ).
trocaLado( dir, esq ).

oper( passa( 2, 0 ) ).
oper( passa( 1, 0 ) ).
oper( passa( 1, 1 ) ).
oper( passa( 0, 1 ) ).
oper( passa( 0, 2 ) ).

solucao( estado( dir, 3, 3 ) ).

pertence( X, [X|_] ).
pertence( X, [_|T] ) :- pertence( X, T ).
```

Como se vê a filosofia das regras mantém-se, tendo-se agora o cuidado de quando se chama *resolver* recursivamente acrescentar EstadoNovo à lista dos estados já percorridos. No entanto antes de o fazermos, verificamos se este estado ainda não pertence a esta lista, recorrendo-se para isso ao operador de negação not.

Agora só falta escrever o predicado possível:

```
possivel( passa( Bcan, Bmis ), estado( Lado,
Can, Mis ),
 estado( LadoF, CanF, MisF ) ) :-
 Can >= Bcan,
 Mis >= Bmis,
```

```
CanF is 3 - Can + Bcan,
MisF is 3 - Mis + Bmis,
bomLado( CanF, MisF ),
DirCanF is 3 - CanF, DirMisF is 3 - MisF,
bomLado( DirCanF, DirMisF ),
trocaLado( Lado, LadoF ).
```

Começa por verificar-se se o número de missionários e canibais na margem Lado é maior que os que se pretendem passar para a outra margem. Depois calcula-se o número de missionários e canibais que vão ficar na outra margem após a viagem. Note-se que os cálculos aritméticos são feitos em Prolog com o auxílio do operador is, que além disso se encarrega de atribuir o resultado à variável da esquerda. Finalmente o número de missionários e canibais que se encontram em cada uma das margens após a travessia, tem que verificar as condições do problema. Mas para isso já nós temos o predicado *bomLado. trocaLado* limita-se a dizer que se estou na esquerda então passo para a direita e vice-versa:

```
trocaLado( esq, dir ).
trocaLado( dir, esq ).
```

Para facilitar a consulta, vamos ainda escrever uma regra auxiliar:

```
resolvo( Estado, Caminho ) :-
  resolver( Estado, Caminho, [ Estado ] ).
```

E pronto. Com esta base de dados o Prolog está apto a resolver o problema bastando para tal lançar o golo:

```
? - resolvo( estado( esq, 3, 3 ), Viagens ).
```

Note-se que existem quatro soluções distintas para este problema. O Prolog dá-as todas mas repete algumas e isto tem que ver com a forma de resolução que utiliza. Num próximo artigo veremos não só como isto acontece mas ainda uma forma de o evitar.

AO CUIDADO DOS MAIS CÉPTICOS

É nesta altura que o leitor não iniciado em IA faz aquela pergunta fatal: Mas então é para isto que serve a IA ? Ou mais sarcástico ainda: Mas então é isto a IA?

A resposta é não. O que aqui se fez foi utilizar um método de IA , a pesquisa não informada, na resolução de um pequeno e inofensivo problema. Este tipo de pesquisa é completamente obsoleto na resolução de problemas minimamente interessantes que tornam aliás obrigatório o recurso às chamadas heurísticas. A ideia das heurísticas é levarem-nos a pesquisar não todo o espaço de alternativas possíveis que se nos deparam, mas apenas aquelas que através de algum critério se nos afiguram como mais prometedoras.

Para finalizar diga-se ainda que a pesquisa informada é apenas um dos numerosos métodos que actualmente se utilizam em IA.

Espero com este artigo ter conseguido abrir o apetite informático do leitor pela aprendizagem do Prolog e da sua aplicação em problemas de IA ou outros.

João Cardoso

O MSDOS POR DENTRO

(PARTE II)

No último artigo foi dada uma primeira abordagem aos Sistemas Operativos referindo o chamado modelo por camadas. Foi também mencioada a maneira de como se processa a comunicação entre todas as camadas.
Neste segundo artigo vamos fazer uma primeira abordagem dos interrupts e funções do DOS, passando posteriormente à prática mostrando como se podem usar todas as funções do MS-DOS nos seus programas de Assembler, C ou até em algumas versões de Basic.
Antes de avançarmos mais nesta matéria torna-se pertinente fazer de seguida uma pequena introdução destinada àqueles que não conhecem nem a lingugem C nem a linguagem Assembler.

INTRODUÇÃO

Como referi no primeiro artigo a linguagem Assembler vai ser usada o menos possível recorrendo-se sempre à linguagem C que se torna de mais fácil entendimento para todos os leitores.

No entanto, vou começar por referir a forma como todas as funções do DOS estudadas vão ser apresentadas, e ajudar aqueles que não conhecem esta linguagem de programação.

Todas as rotinas que apresentarei tomaram a forma duma função que o leitor pode utilizar em todos os seus programas sendo explicados todos os parâmetros de entrada e saída caso existam. Uma função em C representa uma porção de programa que executa uma tarefa que se deseja o mais independente possível do resto do programa. As funções podem ser chamadas em qualquer sítio do programa, até dentro delas próprias...

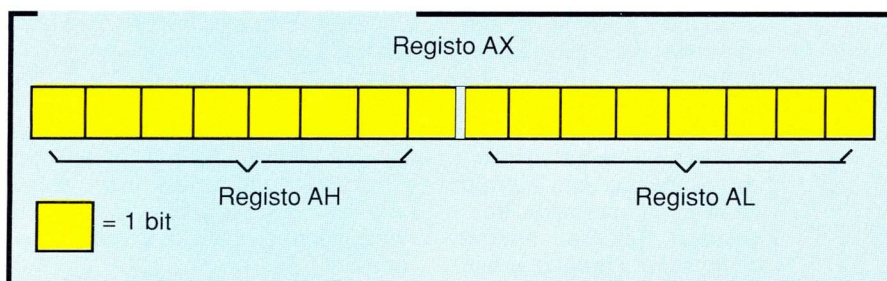
Passando agora à parte "quente", mencionei anteriormente que a comunicação entre as camadas se faz através de interrupts; no caso do utilizador querer utilizar as rotinas do DOS tem exactamente que fazer o mesmo. Assim além de se chamar o interrupt desejado em certos casos deve passar-se em certos registos internos do computador alguns parâmetros para uma plena identificação da acção que se deseja efectuar. Isto pode parecer à primeira vista apenas adequado para ser usado em linguagem Assembler, no entanto, como os programadores de linguagens de alto nível se foram tornando mais exigentes, muitos fabricantes de compiladores foram "obrigados" a incluir nos seus produtos ferramentas capazes de utilizarem os registos do microprocessador.

Neste facto distinguiu-se o C pois actualmente todas as versões de compiladores existentes utilizam esta facilidade de maneira bastante simples.

Refira-se que todos os exemplos aqui apresentados foram usados com o compilador TURBO C — versão 2.0, da Borland, no entanto quem tiver outro compilador não deve ter nenhum género de problema em usar estas rotinas, não sendo necessárias nenhumas alterações em quase todos os compiladores (de facto, não conheço nenhum compilador onde estas rotinas não corram).

Para se aceder aos registos do microprocessador vou utilizar estrutura que está declarada no ficheiro **dos.h** (no turbo c) que está definida da seguinte maneira:

```
struct WORDREGS {
        unsigned int ax, bx, cx, dx, si, di cflag,
falgs;
};
struct BYTEREGS {
        unsigned char al, ah, bl, bh, cl, ch, dl, dh;
};
unionREGS      {
        struct WORDREGS x;
        struct BYTEREGS h;
};
```

A estrutura WORDREGS designa todos os registos que têm a dimensão de 2 bytes enquanto que BYTEREGS designa os que têm um byte, como é óbvio. Note-se que os registos de 2 bytes acabam pela letra "x" e são constituídos por outros 2 registos de 1 byte cada um com as terminações de "h" e "l" conforme sejam os registos de ordem mais ou menos significativa, respectivamente. Para quem não conseguiu compreender esta afirmação está exemplificada em baixo a constituição do registo ax:

Como se pode ver, se o registo ax contiver 5432H, o registo ah contém 54H e o al contém 32H. Já agora refira-se que os campos cflag e flags representam sinalizadores que na altura devida direi a sua utilidade.

Vem esta explicação a propósito de que na apresentação das estruturas atrás feitas figura uma estrutura (REGS) que prevê este mesmo caso. Para através desta estrutura se afectar um registo de 8 bits, o dl por exemplo, utiliza-se a seguinte instrução:

```
reg.h.dl = <valor de 8 bits>
```

sendo reg uma variável que foi declarada como sendo do tipo REGS:

```
union REGS reg;
```

De maneira análoga, para se aceder a um registo de 16 bits (cx por exemplo) usa-se a instrução:

```
reg.x.dx = <valor de 16 bits>
```

Para tornar isto tudo mais claro apresenta-se em seguida um pedaço de código em Assembler e a sua correspondência em C, acrescentando apenas que em C um número em hexadecimal é precedido pelos caracteres "0x" e em Assembler tem o caracter "h" no fim:

ASSEMBLER	C
MOV AL, 41H	reg.h.al=0x41
MOV AH,14	reg.h.al=14
MOV BX,0	reg.x.bx=0

Depois desta pequena introdução teórica vamos passar ao principal motivo deste artigo: O DOS.

INTERRUPTS DO MS-DOS

Tal como para o caso da BIOS o DOS dispõe de alguns interrupts por ele controlados que passo a enunciar:

INT 20 h Interrupt de Fim de Programa — Termina a execução do programa corrente quando é chamado e retorna para o DOS.

Apenas para exemplificar como se poderia chamar este interrupt a partir do C segue-se uma função exemplificativa;

```
sair( )
{ union REGS reg;
  puts ("Voltando ao DOS");
  int86(0x20,&reg,&reg);
  puts("Esta mensagem nunca
  aparece...");
}
```

A função int86 atrás mostrada é a maneira que o C tem de provocar interrupts. O primeiro parâmetro desta função é o número do interrupt, o segundo é o endereço (que se representa por "&") da estrutura que contém o valor dos registos antes de ser executado o interrupt e finalmente o terceiro representa o endereço da estrutura com o valor dos registos após a execução do interrupt.

Este interrupt deve no entanto evitar-se em qualquer linguagem de alto nível podendo conduzir a um crash do computador se não for bem executado. E só é aconselhado para os programadores de Assembler. Mais adiante referirei outra maneira de fazer o mesmo mas com mais segurança.

INT 21h **Rotinas de serviço do DOS** — Este interrupt contém quase todas as funções do DOS de utilidade para o utilizador, facto pelo qual abordarei este interrupt muito detalhadamente mais adiante.

INT 22h **Endereço de fim de programas** — Contém o endereço da rotina para a qual é devolvido controlo após o fim do programa actual. Através deste interrupt pode provocar-se que ao terminar a execução dum determinado programa em vez de se voltar ao interpretador de linha do DOS se possa em vez disso correr um programa elaborado pelo utilizador. Mais uma vez, dados os problemas que o uso indevido deste interrupt pode provocar não nos alongaremos mais e passamos ao próximo interrupt que é:

INT 23h **Endereço de "Break"** — Não, não é break dance, é sim o interrupt que contém o endereço da rotina de tratamento da sequência do teclado "Control-Break" já nossa conhecida (muito conhecida, infelizmente). Alterando esta rotina de interrupt pode fazer-se um tratamento de erros mais razoável doque o DOS costuma fazer, por exemplo chamando uma rotina para confirmar esta acção, evitando assim enganos por vezes fatais. Este é mais um interrupt que apenas deve ser utilizado por programadores com experiência em Assembler.

INT 24h **Tratamento de erros críticos** — Este interrupt é chamado sempre que ocorre um erro no hardware do sistema tal como no controlador de disco, por exemplo. Antes do chamamento deste interrupt deve passar-se como argumento num dos registos do microprocessador qual o tipo de erro. Pode assim simular-se um erro do hardware dentro de qualquer programa bastando evocar este interrupt (embora não esteja a ver qual a utilidade).

INT 25h **Interrupt de escrita no disco (ou disquete)** — Este é um interrupt com alguma importância dado o seu frequente uso quando se quer um "controlo" sobre o conteúdo físico duma disquete. Como se pode ver pelo nome, estão aqui implementadas as rotinas de escrita de dados quer seja numa disquete ou num disco rígido. Provavelmente alguns leito-

res já estão a pensar nas várias maneiras em que este interrupt pode ser usado tanto para o bem, como principalmente para o mal. Deve evitar-se o uso deste interrupt quando não se tem uma noção precisa da organização dos dados dentro duma disquete sob o perigo de poder danificar irrecuperavelmente todo o conteúdo duma disquete (ou disco).

INT 26h **Interrupt de leitura no disco (ou disquete)** — O mesmo que o interrupt anterior mas para leitura. Este sim pode ser utilizado à vontade pelo utilizador pois pode ser bastante útil nas mais variadas situações.

INT 27h **Terminar e manter programas residentes** — Este interrupt é de vital importância pois quando chamado termina a execução do programa actual e instala-o total ou parcialmente em memória de maneira que possa ser chamado em qualquer altura. Os programas que utilizam esta facilidade em geral depois de instalados ficam à espera que uma dada sequência de teclas seja premida. Para se efectuar uma aplicação deste tipo têm que captar obrigatoriamente captar as teclas através do interrupt do teclado (interrupt 9).
Note-se que este interrupt praticamente só foi intensamente usado nas versões 1.x do DOS sendo depois praticamente abandonado em detrimento de outra função que mais adiante referiremos. Este facto deve-se entre outros a que os programas ao tornarem-se residentes só podem ocupar um máximo de 64K o que se foi tornando cada vez mais insuficiente para a maioria das aplicações.

INT 28h **Interrupt de Segurança** — Ora cá está um interrupt bastante misterioso! Como não consegui encontrar em nenhuma documentação a que tivesse acesso uma descrição do funcionamento deste interrupt, tive que usar um método empírico para descobrir o que ele fazia — Dissassemblar toda a rotina de interrupt! Mas, surpresa das surpresas, a rotina não fazia exactamente nada! Felizmente, depois de muito desespero descobri que esta rotina é chamada depois de uma parte crítica do DOS (tal como o acesso a disquetes, por exemplo) ser executada. A utilidade então deste interrupt é o de fornecer uma maneira de o programa a ser instalado não poder interferir com essas zonas críticas, colocando a rotina de activação do programa neste interrupt.

INT 2Fh **Interrupt de Multiplexação** — Descanse o leitor que não vou sequer tentar explicar como funciona este interrupt dada a sua enorme complexidade. Referirei apenas como cultura geral que este interrupt só é usado pelo próprio DOS quando se utilizam os comandos APPEND, ASSIGN, SHARE e PRINT.

AS FUNÇÕES DO MS-DOS

Despois desta explicação que já vai longa de todos os interrupts usados pelo DOS (e de explicar ao leitor que não os deve utilizar!) passemos agora à parte mais útil e afinal o motivo desta série de artigos: as funções do DOS.

Começo por explicar que uma "função do DOS" designa uma das rotinas que é chamada usando o interrupt 21h.

Para aceder a uma dada função deve proceder-se do seguinte modo:

```
reg.h.ah = <número da funçáo desejada>   /* Escolher função */
int86(0x21,&reg,&reg);                   /* Interrupt do DOS */
```

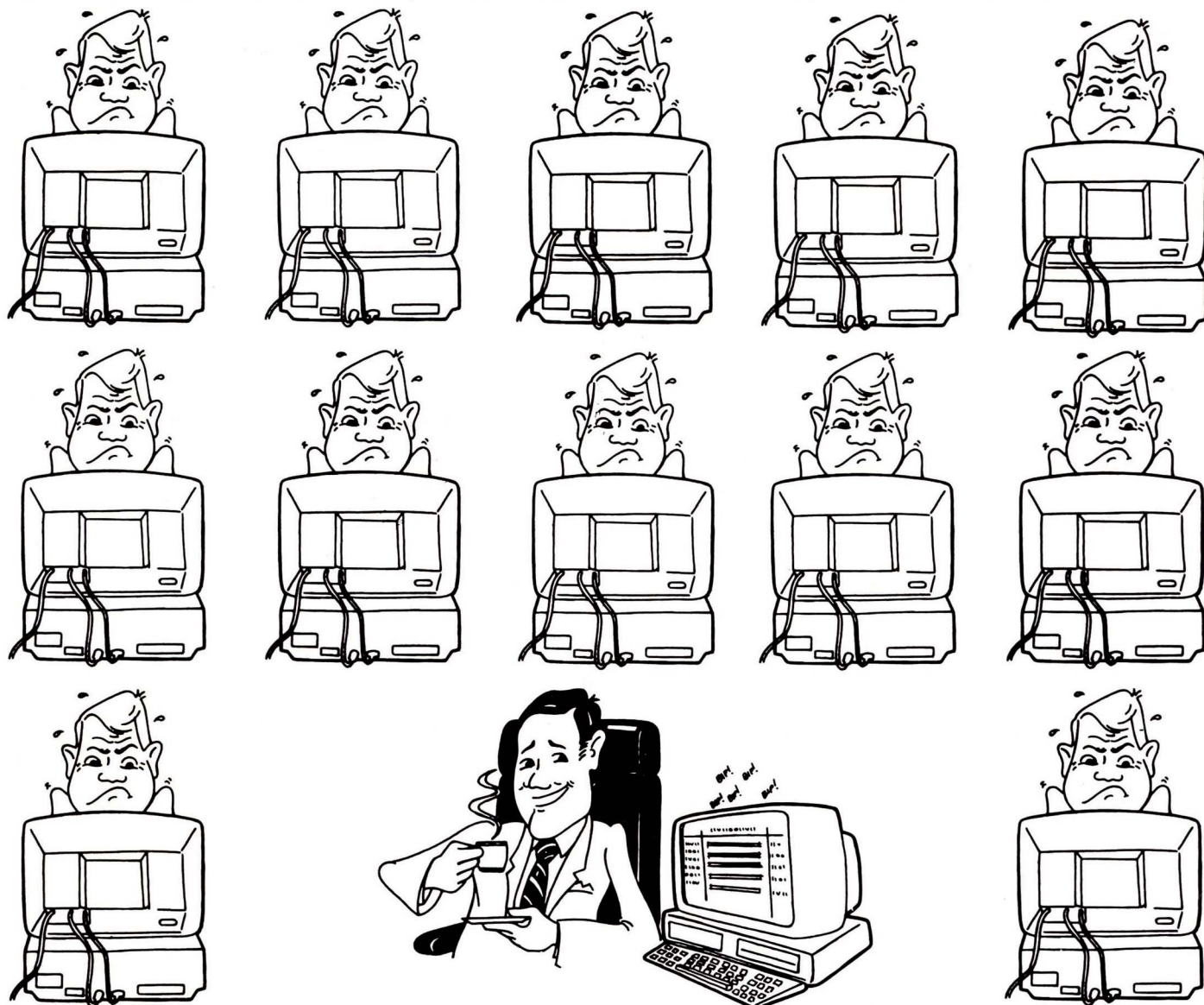
Para se aceder a uma dada função basta assim colocar o número dessa função no registo AH e chamar o interrupt 21h.

CONCLUSÃO

Neste artigo foi feita uma descrição o mais breve possível dos interrupts usados pelo DOS e foi dada a definição de função do DOS. No próximo capítulo vamos referir quais as funções do DOS, qual a sua utilidade e quais os parâmetros. Isto além de incluir pequenos exemplos capazes de serem utilizados e/ou alterados pelos leitores para inclusão nos seus maravilhosos programas "à profissional". Para terminar, lembro que quem tiver alguma dúvida em relação aos temas aqui abordados ou quiser aprofundar alguma parte mencionada apenas ao de leve, escreva uma carta que sempre que tal for pertinente tiraremos todas as dúvidas levantadas.

MIGUEL ÂNGELO VITORINO

Nota: Por lapso na primeira parte deste artigo não foi publicada esta figura, que por várias vezes vem referida no texto. Para os leitores que nos têm seguido desde essa primeira parte do artigo aqui redimimos o nosso erro.

Já pensou quanto vale um minuto?

Um minuto é o espaço de tempo necessário
para evitar a decisão errada.
É tomar a melhor decisão e ser eficaz.
É pensar antes de agir.
É determinar com rigor
as necessidades da sua empresa
na área do software.
Por isso, concebemos os programas adequados
à sua situação específica.
PERDER UM MINUTO É GANHAR... MUITO TEMPO!

A solução

Shopping Cacém, loja 2.42 Telef. 928 09 29 Cacém

mahe

5 MIL, 10 MIL, 20 MIL, E... MUITO MAIS PRÉMIOS

Mais uma vez os leitores começaram a responder da melhor forma a um passatempo/concurso que a AM iniciou há cerca de três meses e que pretende levar a cabo todas as edições.

A iniciativa, que para os leitores mais recentes podemos descrever como um concurso mensal em que se distinguem os três melhores trabalhos enviados para esta secção (programas, rotinas, "dicas", truques, artigos técnicos, etc.) durante o mês imediatamente anterior à publicação da AM, aparentemente interessou a muitos leitores que não quiseram deixar de partilhar os seus conhecimentos com os restantes utilizadores, arriscando-se a ganhar 20 mil escudos em compras nas lojas Socartel. No momento em que fechamos esta edição continuamos a receber trabalhos muito bons (não obrigatoriamente extensos), que, como é lógico, serão analisados no próximo número obtendo, ou não, um dos prémios que estão à sua disposição.

Nesta AM os primeiros três premiados estão correctamente identificados nos trabalhos que publicamos (curiosamente, o terceiro prémio é atribuido a dois utilizadores que se juntaram para apresentar o trabalho em causa). Todos os outros trabalhos recebidos até iniciarmos a preparação deste número, e não incluidos nos três primeiros, serão gradualmente publicados estando então automaticamente "inscritos" para uma enorme atribuição de prémios a fazer em Dezembro deste ano. O mesmo se irá passar durante todos os meses que nos esperam até lá, e em relação a todos os trabalhos que entretanto recebermos.

Para que não existam quaisquer dúvidas, para além dos premiados os leitores considerados neste número foram os seguintes:

—Paulo Jorge — Silves
—Carlos Alberto E.R. Paredes — Coimbra
—Eurico Borges — Matosinhos
—Jorge Manuel Lourenço Ramos — Miranda do Corvo
—Ricardo Pereira — Maia
—João Beato
—Pedro Lúcio — Oeiras

1º PRÉMIO

JOAQUIM NORBERTO CARDOSO PIRES SILVA
COIMBRA

Sou um leitor habitual da AM, revista que leio com enorme prazer, embora não entenda bem a vossa preferência pela linguagem BASIC. Na verdade, as linguagens de programação C, TURBO-PASCAL e FORTRAN são as mais utilizadas em ciência sendo também as mais versáteis e poderosas. Proponho a criação de secções dedicadas a estas linguagens na AM. Eu próprio possuo razoável experiência nessas linguagens (especialmente em TURBO-PASCAL), pelo que vos envio alguns programas da minha autoria que poderão constituir um incentivo para os leitores da AM iniciarem o estudo destas linguagens.

O primeiro programa constitui um pequeno (des)codificador de mensagens, que pode ser utilizado pelos mais espertos espiões, bem como pelos mais inofensivos pares amorosos.

O segundo programa não é um programa, mas sim uma UNIT, e é dedicada aos utilizadores do TURBO-PASCAL 4.0 ou 5.0. Esta UNIT contém dois procedimentos úteis:

PROCEDIMENTO ESPERA: este procedimento melhora o procedimento DELAY, da UNIT CRT, que vem no TURBO-PASCAL aumentando a sua precisão. O procedimento DELAY varia bastante com o computador utilizado, sendo bastante aborrecido quando precisam os de tempos de espera constantes. O procedimento ESPERA tem uma precisão de 1/18 segundos e utiliza um parâmetro real, o que é mais conveniente do que os milisegundos exigidos pelo DELAY.

O procedimento proposto faz leitura do relógio do PC utilizando a função $2C do DOS int 21H. Este serviço guarda a informação nas seguintes variáveis/registos:

CH -> horas (0 a 23)
CL -> minutos (0 a 59)
DH -> segundos (0 a 59)
DL -> centésimos de segundo (0 a 99).

PROCEDIMENTO PASSWORD: este procedimento permite-lhe proteger os seus programas (ver exemplo no programa (des)codificador).

O compilador que utilizei foi o TURBO-PASCAL 5.0 da BORLAND.

Devo referir por último que encontrei um procedimento parecido ao procedimento ESPERA numa revista norte-americana (PC MAGAZINE).

```
unit wait;
interface
uses dos,crt;
procedure espera(nseg:real);
procedure password(p,s,t,q,n:integer);
implementation
procedure password(p,s,t,q,n:integer);
var i,x,a,b,c,d,e:integer;
    control:boolean;
begin
      randomize;
      repeat
              clrscr;
              writeln('(c) Norberto Pires .');
              x:=trunc(random*10000);
              write('PASSWORD(',x,'):');
              a:=x mod 10;b:=(x div 10) mod 10;c:=(x div 100) mod 10;
              d:=(x div 1000) mod 10;e:=x div 10000;
              read(i);if i=(p*a+s*b+t*c+q*d+n*e) then control:=true;
      until control=true;
end;
procedure espera(nseg:real);
const
      seg_por_dia=86400.0;         {60*60*24}
var
   segpar,seg:real;
   tempopedido:boolean;
procedure ler_relogio(var seg:real);
const
      seg_por_hora=3600.0;
      seg_por_minuto=60.0;
var
   registo:registers;
begin
      registo.AH:=$2C;msdos(registo);
      seg:=seg_por_hora*(registo.CH)+seg_por_dia*(registo.CL)+
      0.01*(registo.DL)+registo.DH;
end;
begin
ler_relogio(segpar);
repeat
      ler_relogio(seg);
      if seg-segpar>=0.0 then tempopedido:=seg-segpar>=nseg
      else tempopedido:=seg_por_dia-segpar+seg>=nseg;
until tempopedido;
end;
end.
```

```
program codigo(origem,destino,input,output);
uses crt,dos,wait ;
label 1;
type dominio=1..5;
var origem,destino:text;
    mes,dia,ano,s:word;
    a,b,c,w,x,p,i,m:integer;
    D:array [1..5] of integer;
    numero:dominio;
    cr,ch:char;
    control:boolean;
procedure clear( v:integer);  {  Este procedimento permite uma maior }
var  i:integer;               {   clareza do ecran .                 }
begin
     i:=0;
     repeat
     writeln;i:=i+1;
     until i=v;
end;
procedure copialinha(var f:text;v:integer);
var o:char;
    g:text;            {  Este procedimento  copia para o ficheiro origem.txt }
begin                   {ou para o ficheiro destino.txt uma linha escrita  }
     begin              {atraves do teclado , caso esta seja para codificar }
           while not eoln(f) do     {ou para descodificar , respectivamente.}
           begin
                 read(f,o);if v=1 then write(origem,o);if v=2 then
                 write(destino,o);
           end;
                 readln(f);if v=1 then writeln(origem);if v=2 then
                 writeln(destino);
     end;
end;
procedure erro(numero:dominio);      {Mensagens de erro.}
begin
     write('***  ');
     case numero of
     1:writeln('ERRO: Ficheiro acabado prematuramente .   ***');
     2:writeln('ERRO: Deve introduzir um numero entre "0" e maxint .  ***');
     end;
     writeln('                    ***   Programa terminado  ***');
     control:=false;a:=5;
end;
begin
      highvideo;password(1,1,1,1,1);lowvideo;
assign(origem,'origem.txt');assign(destino,'destino.txt');
repeat
      control:=true;
      getdate(ano,mes,dia,s);
      clrscr;highvideo;
      clear(3);write('                    ##### PROGRAMA DE (DES)CODIFI');
      writeln(chr(128),chr(142),'O #####');
      clear(2);write('                              M',chr(136),'s:',mes);
      writeln('  Dia:',dia,'  Ano:',ano);
      clear(3);writeln('MENU PRINCIPAL');lowvideo;
      clear(2);writeln('1-> Escrever texto.');writeln('2-> Codificar texto.');
      writeln('3-> Descodificar texto.');writeln('4-> Terminar.');
      clear(5);write('####->');read(a);
      if a=1 then
      1:begin
         clrscr;clear(2);
         write('PARA CODIFICAR OU PARA DESCODIFICAR(1/2) ?');read(w);
         if (w<>1) and(w<>2) then goto 1;
         if w=1 then
         begin
             rewrite(origem);
             writeln('#-> CODIFICAR.');
         end;
         if w=2 then
         begin
              rewrite(destino);
              writeln('#-> DESCODIFICAR.')
         end;
write('Numero de linhas ->');read(b);clear(2);i:=0;
while not eof(input) and (i<=b) do
begin                                {Copiar o texto para um dos ficheiros.}
    copialinha(input,w);i:=i+1;
end;
if i<b then erro(1);if w=1 then close(origem);if w=2 then close(destino);
readln;
end;
if (a=2) or (a=3) then
begin
     clrscr;clear(2);
     write('## CHAVE CODIGO ##->');readln(c);
     if(c>maxint) or(c<0) then erro(2) else
           begin
           i:=1;                              {Analise da chave de codigo.}
           repeat
           D[i]:=c mod 10;c:=c div 10 ;i:=i+1;
           until i=6;
           m:=5;
           if (D[5]=0) then m:=4;
           if (D[5]=0) and (D[4]=0) then m:=3;
           if (D[5]=0) and (D[4]=0) and (D[3]=0) then m:=2;
           if (D[5]=0) and (D[4]=0) and (D[3]=0) and (D[2]=0) then m:=1;
end;
if a=2 then
begin
reset(origem);
rewrite(destino);i:=1;
while not eof(origem) do
begin
    while not eoln(origem) do
    begin
    read(origem,cr);
    if cr in(['A'..'Z']+['a'..'z']+['0'..'9']) then
    begin
        ch:=chr(ord(cr)+D[i]);
        if (cr in ['A'..'Z']) and ((ord(cr)+D[i]-1)>ord('Z')) then
        ch:=chr(ord(cr)+D[i]-ord('Z')+ord('A')-1);
        if (cr in ['a'..'z']) and ((ord(cr)+D[i])>ord('z')) then
        ch:=chr(ord(cr)+D[i]-ord('z')+ord('a')-1);
        if (cr in ['0'..'9']) and ((ord(cr)+D[i])>ord('9')) then
        ch:=chr(ord(cr)+D[i]+ord('0')-ord('9')-1);
        write(destino,ch);write(ch);                {CODIFICACAO.}
   end
   else begin write(destino,cr);write(cr);end;
   i:=i+1;if i=m+1 then i:=1;
   end;
```

```
      readln(origem);writeln(destino);writeln;
   end;close(origem);close(destino);clear(3);
   write('Prima uma tecla .');readln;control:=true;
   end;
   if a=3 then
   begin
        reset(destino);
        rewrite(origem);i:=1;
        while not eof(destino) do
        begin
             while not eoln(destino) do
             begin
             read(destino,cr);
             if cr in(['A'..'Z']+['a'..'z']+['0'..'9']) then
             begin
                  ch:=chr(ord(cr)-D[i]);
                  if (cr in ['A'..'Z']) and ((ord(cr)-D[i])<ord('A')) then
                  ch:=chr(ord('Z')-(ord('A')-ord(cr)+D[i])+1);
                  if (cr in ['a'..'z']) and ((ord(cr)-D[i])<ord('a')) then
                  ch:=chr(ord('z')-(ord('a')+D[i]-ord(cr))+1);
                  if (cr in ['0'..'9']) and((ord(cr)- D[i])<ord('0')) then
                  ch:=chr(ord('9')-(ord('0')-ord(cr)+D[i])+1);
                  write(origem,ch);write(ch);
             end                                          {DESCODIFICACAO.}
             else begin write(origem,cr);write(cr);end;
             i:=i+1;if i=m+1 then i:=1;
        end;readln(destino);writeln(origem);writeln;
        end;
        close(origem);close(destino);clear(3);
        write('Prima uma tecla .');readln;control:=true;
        end;
   end;
   if a=4 then
   begin
   highvideo;
   clrscr;clear(2);writeln('(c) Norberto Pires . Coimbra 1989');lowvideo;
   write('-----------------------------------');
   control:=false;
   end;
   until  control=false;
   end.
```

2º PRÉMIO

NUNO FILIPE RIBEIRO FERREIRA (13 anos)

LISBOA

O programa, cuja listagem envio é um ficheiro de clientes para uma empresa fictícia (Fitas & Bolas, Lda.). Este ficheiro tem a função de gerir os clientes da empresa. Utiliza os seguintes campos:

- Código do cliente (3 caracteres)
- Nome do cliente (40 caracteres)
- Morada (40 caracteres)
- Localidade (15 caracteres)
- Nº de contribuinte (9 caracteres)
- Valor da compra.

As opções de manipulação do ficheiro são as seguintes: Inserção, alteração, anulação, consulta, listagem, criação de ficheiro e fim.
A minha sugestão é que cada leitor tente adaptar o programa às suas necessidades.

Nota: *O programa destina-se ao Amstrad CPC.*

```
1 COD=0
3 DIM MATG$(50,6)
10 CLS
20 FOR l=1 TO 20
30 LOCATE 10,1:PRINT "FITAS & BOLAS LDA."
40 NEXT l
50 DEG
60 FOR A=1 TO 360
70 PLOT SIN(A)*50+500,COS(A)*50+300
80 NEXT A
90 PLOT 10,225
100 DRAW 130,360
110 FOR A=1 TO 360
120 PLOT SIN(A)*50+70,COS(A)*50+300
130 NEXT A
140 LOCATE 1,25:PRINT " CARREGUE EM A PARA CONTINUAR "
150 IF INKEY$="A" OR INKEY$="a" THEN 170
160 GOTO 150
170 CLS
180 FOR C=1 TO 18
190 LOCATE C,5:PRINT " **MENU** "
200 NEXT C
210 PLOT 120,300
220 DRAW 550,300
230 DRAW 550,78
240 DRAW 120,78
250 DRAW 120,300
260 LOCATE 9,8:PRINT "1-INSERCCAO DE CLIENTES"
270 LOCATE 9,10:PRINT "2-ALTERACAO DE CLIENTES"
280 LOCATE 9,12:PRINT "3-ANULACAO DE UM CLIENTE"
290 LOCATE 9,14:PRINT "4-CONSULTA"
300 LOCATE 9,16:PRINT "5-LISTAGEM"
310 LOCATE 9,18:PRINT "6-CRIACAO DO FICHEIRO"
320 LOCATE 9,20:PRINT "7-FIM"
330 LOCATE 1,24:PRINT "QUAL A OPCAO "
340 INPUT OP
343 IF OP<1 OR OP>7 THEN CLS:LOCATE 10,10:PRINT "OPCAO INVALIDA":GOTO 170
350 ON OP GOSUB 440,730,1020,1200,1470,3000,4000
400 CLS
410 GOTO 170
440 CLS
442 IF COD=0 THEN FOR L=1 TO 40:LOCATE 1,10:PRINT "ANTES DE FAZER QUAISQUER OPER
ACOES COM O FICHEIRO DEVE CRIA-LO":NEXT L:GOTO 170
445 A=1
460 OPENIN "CLIENTES"
465 FOR X=1 TO NREGISTO
470 INPUT #9,MATG$(A,1),MATG$(A,2),MATG$(A,3),MATG$(A,5),MATG$(A,6)
480 A=A+1
490 NEXT X
500 CLOSEIN
510 INPUT "QUANTOS REGos. QUER INSERIR ";QREG
520 FOR CIC=A+1 TO A+QREG
525 NREGISTO=NREGISTO+1
530 INPUT "CODIGO DO CLIENTE ";MATG$(CIC,1)
540 IF LEN (MATG$(CIC,1))>3 THEN GOTO 530
550 CLS
560 INPUT "NOME DO CLIENTE ";MATG$(CIC,2)
570 IF LEN (MATG$(CIC,2))>40 THEN 560
580 CLS
590 INPUT "MORADA DO CLIENTE ";MATG$(CIC,3)
600 IF LEN(MATG$(CIC,3))>40 THEN GOTO 590
605 CLS
610 INPUT "LOCALIDADE ";MATG$(CIC,4)
620 IF LEN (MATG$(CIC,4))>15 THEN GOTO 610
625 CLS
630 INPUT "No.DE CONTRIBUINTE ";MATG$(CIC.5)
640 IF LEN (MATG$(CIC,5))>9 THEN GOTO 630
650 CLS
660 INPUT "VALOR DA COMPRA ";VALCOMP$
670 CLS:NEXT CIC
680 OPENOUT "CLIENTES"
690 FOR X=1 TO A+QREG
700 WRITE #9,MATG$(X,1),MATG$(X,2),MATG$(X,3),MATG$(X,4),MATG$(X,5),MATG$(X,6)
710 NEXT X
715 CLOSEOUT
720 RETURN
730 A=1
740 OPENIN "CLIENTES"
745 WHILE NOT EOF
750 INPUT #9,MATG$(A,1),MATG$(A,2),MATG$(A,3),MATG$(A,4),MATG$(A,5),MATG$(A,6)
760 A=A+1
770 WEND
775 CLOSEIN
780 CLS:INPUT "NOME DO CLIENTE CUJA FICHA PRETENDE ALTERAR ";NCFA$
790 FOR L=1 TO A
800 IF NCFA$=MATG$(L,2) THEN GOTO 835
810 NEXT L
820 PRINT "O NOME QUE PEDIU NAO EXISTE ":GOTO 960
835 CLS:LOCATE 1,25:PRINT "(ENTER) P/ O PROXIMO CAMPO"
840 LOCATE 2,2:PRINT "CODIGO :";MATG$(L,1)
850 LOCATE 8,2:INPUT MATG$(L,1):IF LEN(MATG$(L,1))>3 THEN LOCATE 1,20:PRINT "COD
IGO GRANDE DEMAIS":GOTO 840
860 LOCATE 2,4:PRINT "NOME :";MATG$(L,2)
870 LOCATE 6,4:INPUT MATG$(L,2):IF LEN(MATG$(L,2))>40 THEN CLS:PRINT "NOME GRAND
E DEMAIS":GOTO 860
880 LOCATE 2,6:PRINT "MORADA :";MATG$(L,3)
890 LOCATE 8,6:INPUT MATG$(L,3):IF LEN(MATG$(L,3))>40 THEN CLS:PRINT "MORADA GRA
NDE DEMAIS":GOTO 880
900 LOCATE 2,8:PRINT "LOCALIDADE :";MATG$(L,4)
910 LOCATE 12,8:INPUT MATG$(L,4):IF LEN(MATG$(L,4))>15 THEN CLS:PRINT "LOCALIDAD
E GRANDE DEMAIS":GOTO 900
920 LOCATE 2,10:PRINT "No. DE CONTRIBUINTE :";MATG$(L,5)
930 LOCATE 21,10:INPUT MATG$(L,5):IF LEN(MATG$(L,5))>9 THEN CLS:PRINT "No. DE CO
NTRIBUINTE GRANDE DEMAIS":GOTO 920
940 LOCATE 2,12:PRINT "VALOR DA COMPRA :";MATG$(L,6)
950 LOCATE 17,12:INPUT MATG$(L,6)
960 OPENOUT "CLIENTES"
970 FOR L=1 TO A
980 WRITE #9,MATG$(L,1),MATG$(L,2),MATG$(L,3),MATG$(L,4),MATG$(L,5),MATG$(L,6)
990 NEXT L
1000 CLOSEOUT
1005 INPUT "QUER CONTINUAR [S/N] ";ONSN$
1010 IF ONSN$="S" OR ONSN$="s" THEN 780
1015 IF ONSN$="N" OR ONSN$="n" THEN CLS:RETURN
1017 GOTO 1005
1020 CLS
1025 A=1
1030 OPENIN "CLIENTES"
1040 INPUT #9,MATG$(A,1),MATG$(A,2),MATG$(A,3),MATG$(A,4),MATG$(A,5),MATG$(A,6)
1050 A=A+1
1060 CLOSEIN
1070 INPUT "NOME DO CLIENTE A ANULAR";CLIANUL$
1080 FOR D=1 TO A
1090 IF MATG$(D,1)=CLIANUL$ THEN MATG$(D,1)=" ":MATG$(D,2)=" ":MATG$(D,3)=" ":MA
TG$(D,4)=" ":MATG$(D,5)=" ":MATG$(D,6)=" ":NREGISTO=
NREGISTO-1:GOTO 1120
1100 NEXT D
1110 PRINT "ESSE NOME NAO EXISTE":GOTO 1170
1120 OPENOUT "CLIENTES"
1130 FOR D=1 TO A
1140 WRITE #9,MATG$(D,1),MATG$(D,2),MATG$(D,3),MATG$(D,4),MATG$(D,5),MATG$(D,6)
1150 NEXT D
1160 CLOSEOUT
1170 INPUT "ANULAR OUTRO (S/N) ";ANULOUT$
1180 IF ANULOUT$="S" OR ANULOUT$="s" THEN CLS:GOTO 1070
1190 IF ANULOUT$="N" OR ANULOUT$="n" THEN RETURN
1195 GOTO 1170
1200 CLS
1205 A=1
1210 OPENIN "CLIENTES"
1220 WHILE NOT EOF
1230 INPUT #9,MATG$(A,1),MATG$(A,2),MATG$(A,3),MATG$(A,4),MATG$(A,5),MATG$(A,6)
1240 A=A+1
1250 WEND
1260 CLOSEIN
1270 INPUT "NOME DO CLIENTE A CONSULTAR ";CLICONSULT$
1280 FOR D=1 TO A
1290 IF CLICONSULT$=MATG$(D,2) THEN 1330
1300 NEXT D
1310 PRINT "ESSE CLIENTE NAO EXISTE!":GOTO 1380
1330 LOCATE 2,2:PRINT "CODIGO : ";MATG$(D,1)
1340 LOCATE 2,4:PRINT "NOME : ";MATG$(D,2)
1350 LOCATE 2,6:PRINT "MORADA : ";MATG$(D,3)
1360 LOCATE 2,8:PRINT "LOCALIDADE : ";MATG$(D,4)
1365 LOCATE 2,10:PRINT "No. DE CONTRIBUINTE : ";MATG$(D,5)
1370 LOCATE 2,12:PRINT "VALOR DA COMPRA : ";MATG$(D,6)
1380 OPENOUT "CLIENTES"
1390 FOR D=1 TO A
1400 WRITE #9,MATG$(D,1),MATG$(D,2),MATG$(D,3),MATG$(D,4),MATG$(D,5),MATG$(D,6)
1410 NEXT D
1420 CLOSEOUT
1430 INPUT "QUER OUTRO [S/N] ";SIMOUNAO$
1440 IF SIMOUNAO$="S" OR SIMOUNAO$="s" THEN CLS:GOTO 1270
1450 IF SIMOUNAO$="N" OR SIMOUNAO$="n" THEN RETURN
1460 GOTO 1430
1470 CLS
1475 A=1
1480 OPENIN "CLIENTES"
1490 WHILE NOT EOF
1500 INPUT #9,MATG$(A,1),MATG$(A,2),MATG$(A,3),MATG$(A,4),MATG$(A,5),MATG$(A,6)
1510 A=A+1
1520 WEND
1530 CLOSEIN
1540 LOCATE 10,3:PRINT "** LISTAGEM POR CODIGO **"
1550 FOR L=2 TO A
1560 FOR D=1 TO A-1
1570 IF MATG$(L,1)<MATG$(D,1) THEN 1600
1580 NEXT D,L
1590 GOTO 1790
1600 AUX$=MATG$(L,1)
1610 AUXI$=MATG$(L,2)
1620 AUXIL$=MATG$(L,3)
1630 AUXILI$=MATG$(L,4)
1640 AUXILIA$=MATG$(L,5)
1650 AUXILIAR$=MATG$(L,6)
1660 MATG$(L,1)=MATG$(D,1)
1670 MATG$(L,2)=MATG$(D,2)
1680 MATG$(L,3)=MATG$(D,3)
1690 MATG$(L,4)=MATG$(D,4)
1700 MATG$(L,5)=MATG$(D,5)
1710 MATG$(L,6)=MATG$(D,6)
1720 MATG$(D,1)=AUX$
1730 MATG$(D,2)=AUXI$
1740 MATG$(D,3)=AUXIL$
1750 MATG$(D,4)=AUXILI$
1760 MATG$(D,5)=AUXILIA$
1770 MATG$(D,6)=AUXILIAR$
1780 GOTO 1570
1790 NREGISTO=0
1800 FOR L=1 TO A
1810 NREGISTO=NREGISTO+1
1820 LOCATE 10,5:PRINT "REGISTO No.-";NREGISTO
1830 LOCATE 2,7:PRINT "CODIGO :";MATG$(L,1)
1840 LOCATE 2,9:PRINT "NOME :";MATG$(L,2)
1850 LOCATE 2,11:PRINT "MORADA :";MATG$(L,3)
1860 LOCATE 2,13:PRINT "LOCALIDADE :";MATG$(L,4)
1870 LOCATE 2,15:PRINT "No. DE CONTRIBUINTE ";MATG$(L,5)
1880 LOCATE 2,17:PRINT "VALOR DA COMPRA ";MATG$(L,6)
1890 LOCATE 1,24:PRINT "TECLE [P] P/ PROXIMO "
1900 IF INKEY$="P" OR INKEY$="p" THEN GOTO 1920
1910 GOTO 1900
1920 NEXT L
1930 PRINT "LISTAGEM COMPLETA"
1940 OPENOUT "CLIENTES"
1950 FOR L=1 TO A
1960 WRITE #9,MATG$(L,1),MATG$(L,2),MATG$(L,3),MATG$(L,4),MATG$(L,5),MATG$(L,6)
1970 NEXT L
1980 CLOSEOUT
1990 RETURN
3000 CLS
3001 IF COD=1 THEN PRINT "O FICHEIRO JA FOI ABERTO":GOTO 170
3005 COD=1
3010 OPENOUT "CLIENTES"
3015 LET NREGISTO=NREGISTO+1:PRINT "REGISTO No. ";NREGISTO
3030 LOCATE 2,3:INPUT "CODIGO DO CLIENTE";CODCLI$
3040 IF LEN(CODCLI$)>3 THEN PRINT "CODIGO GRANDE DEMAIS":GOTO 3030
3050 CLS
3060 INPUT "NOME DO CLIENTE ";NOMECLI$
3070 IF LEN(NOMECLI$)>40 THEN PRINT "NOME GRANDE DEMAIS ":GOTO 3060
3080 CLS
3090 INPUT "MORADA DO CLIENTE ";MORADACLI$
3100 IF LEN(MORADACLI$)>40 THEN PRINT "MORADA GRANDE DEMAIS":GOTO 3090
3110 CLS
3120 INPUT "LOCALIDADE ";LOCALICLI$
3130 IF LEN(LOCALICLI$)>15 THEN PRINT "LOCALIDADE GRANDE DEMAIS":GOTO 3120
3140 CLS
3150 INPUT "No.DE CONTRIBUINTE ";NCONTRICLI$
3160 IF LEN(NCONTRICLI$)>9 THEN PRINT "No. DE CONTRIBUINTE GRANDE DEMAIS ":GOTO
3150
3170 CLS
3180 INPUT "VALOR DA COMPRA ";VALCOMP$
3190 CLS
3200 WRITE #9,CODCLI$,NOMECLI$,MORADACLI$,LOCALICLI$,NCONTRICLI$,VALCOMP$
3210 INPUT "OUTRO REGISTO (S/N) ";SN$
3220 IF SN$="S" OR SN$="s" THEN 3015
3230 IF SN$<>"N" AND SN$="n" THEN CLS:3210
3240 CLOSEOUT
3250 RETURN
```

3º PRÉMIO

CARLOS JORGE MOUTINHO DE PINA NEVES

MAIA

O meu nome é Carlos Jorge Neves, e sou um apaixonado pelo mundo dos computadores. Juntamente com uma série de amigos, formamos um grupo bastante interessado pelo mundo fantástico da informática.
Assim, gostariamos de participar na vossa revista, dando uma pequena contribuição. Trata-se da solução (uma delas) para o jogo Leisure Suite Larry in the Land of the Lounge Lizards.
O motivo que nos leva a escrever para a vossa revista, é o facto de acharmos que a Amstrad magazine contribui para o crescimento do mercado informático e para o desenvolvimento de novas mentalidades. No entanto o principal motivo será querer ajudar todos aqueles que se encontram desesperados para tentar acabar o referido jogo. Sendo assim, aqui vai a solução:
Depois de entrar no jogo (respondendo a todas as questões CORRECTAMENTE), fazer o seguinte:
Chamar o táxi com TAXI; entrar com GET IN ir até à loja com GO STORE; pagar ao motorista (caso contrário levamos um tiro) com PAY; saír com GET OUT; entrar na loja e junto à primeira prateleira escrever GET MAGAZINE; ir à primeira prateleira (do fundo, junto à parede) e escrever GET WINE; junto ao balcão escrever GET RUBBER; (depois de envergonhado) saír da loja e se aparecer um bebâdo escrever GIVE WINE (caso contrário ele aparecerá mais tarde); TAXI; GET IN; GO BAR; PAY; junto à porta do bar OPEN DOOR; junto à cadeira vazia SIT; WHISKY; GET UP; entrar no corredor (junto à caixa de música) e junto da mesa GET ROSE; junto ao bebâdo GIVE WHISKY; junto à porta OPEN DOOR; na sanita SIT; GET UP; SEE GRAFFITI (várias vezes); junto ao lavatório GET RING; junto à porta OPEN DOOR ; para um pouco de música PLAY MUSIC; junto à porta do lado direito KNOCK; KEN SENT ME; USE REMOTE CONTROL (em frente da televisão); CHANGE CHANNEL (muitas vezes); subir as escadas e junto à mesa GET CANDY BOX; na cama EXAMINE WOMAN; GET CLOTHES OFF; USE RUBBER; o momento porque muitos esperavam FUCK HER (opss!!); depois de uma desilusão com o censurado GET RUBBER OFF; junto à janela OPEN WINDOW; GO OUT THE WINDOW; depois de caír no caixote do lixo andando para o lado esquerdo GET HAMMER; GO OUT; andando para o lado esquerdo e em frente do bar TAXI; GET IN; GO HOTEL; PAY; GO OUT; aparecendo um homem com barril BUY APPLE (senão aparecerá mais tarde); entrar no casino e junto a uma máquina de jackpot PLAY SLOT — aqui temos um pequeno truque, que será gravar o jogo para proteger o dinheiro que temos, só que a gravação será feita na Ramdrive (C), jogar com F4, F6 e F8 e quando o dinheiro estiver "em baixo" recuperar o montante que tinhamos com Restore, caso se ganhe, gravar pelo mesmo processo (eih!; que Tal?), quando estivermos satisfeitos ou se chegar ao limite da máquina, saír — andar até ao final do corredor e junto ao cinzeiro GET CARD; saír do casino e TAXI; GET IN; GO DISCO; PAY; GET OUT; entrar na discoteca mas SHOW CARD primeiro; junto à mesa com a mulher SIT; LOOK WOMAN; GIVE ROSE; GIVE RING; GIVE CANDY (que mulher exigente); DANCE; GET UP; depois de umas risadas valentes voltar à mesa e SIT; LOOK WOMAN; GIVE MONEY (mais?!); GET UP; saír da discoteca e TAXI; GET IN; GO HOTEL; PAY; GO OUT; (se só agora apareceu o homem do barril, compre-lhe a maçã); entrar no casino e jogar novamente jackpot (porque as mulheres são caras e estamos "lisos") com PLAY SLOT (utilize o truque..); depois de estar rico saír do casino, andar para o lado direito e entrar na capela com OPEN DOOR; junto ao padre GET MARRIED; depois de uma desilusão com o padre, voltar ao casino e junto ao elevador PRESS FOUR; junto da porta com o coração KNOCK; junto da "nossa" amada KISS HER; (vinho?!!) junto do rádio TURN RADIO ON; depois de esperar um pouco e de ficar a saber o número de telefone da loja (5558039) saír com OPEN DOOR; junto do elevador PRESS ONE; saír do casino e TAXI; GET IN; GO STORE; PAY; GO OUT; junto do telefone DIAL PHONE; 5558039; WINE; HONEYMOON SUITE; TAXI; GET IN; GO OUT; entrar no casino e no elevador PRESS FOUR; na porta do coração KNOCK; junto da amada TAKE WINE; e FUCK HER; mais uma encrenca mas USE KNIFE resolve o assunto; GET ROPE; junto à porta OPEN DOOR; no elevador PRESS ONE; jogar jackpot, pois o casamento ficou caro e só temos 10 dólares (utilize o truque...); saír do casino e TAXI; GET IN; BAR; PAY; GO OUT; OPEN DOOR; KNOCK; KEN SENT ME; subir escadas e na janela GO OUT THE WINDOW; na varanda TIE ROPE TO LARRY; TIE ROPE TO BALCONY; GET PILLS; USE HAMMER; GET PILLS; PULL ROPE; UNTIE ROPE; caír no caixote e GO OUT; andar para a frente do bar e TAXI; GET IN; GO HOTEL; PAY; GO OUT; entar no casino e no elevador PRESS EIGHT ; junto ao balcão LOOK WOMAN (não é um espanto?) CALL WOMAN (várias vezes); GIVE PILLS TO HER; GO OUT; EXAMINE DESK; PRESS BUTTON; entrar na porta agora aberta, ir para o lado direito da sala (no quarto) e junto à porta OPEN DOOR; SEE; TAKE DOLL (olhem como ele corre); junto à mulher GO BATH; LOOK WOMAN (uau! mama mia...) CALL WOMAN (várias vezes); GIVE APPLE (Larry deixou de ser virgem e o fogo de artifício anima a festa).

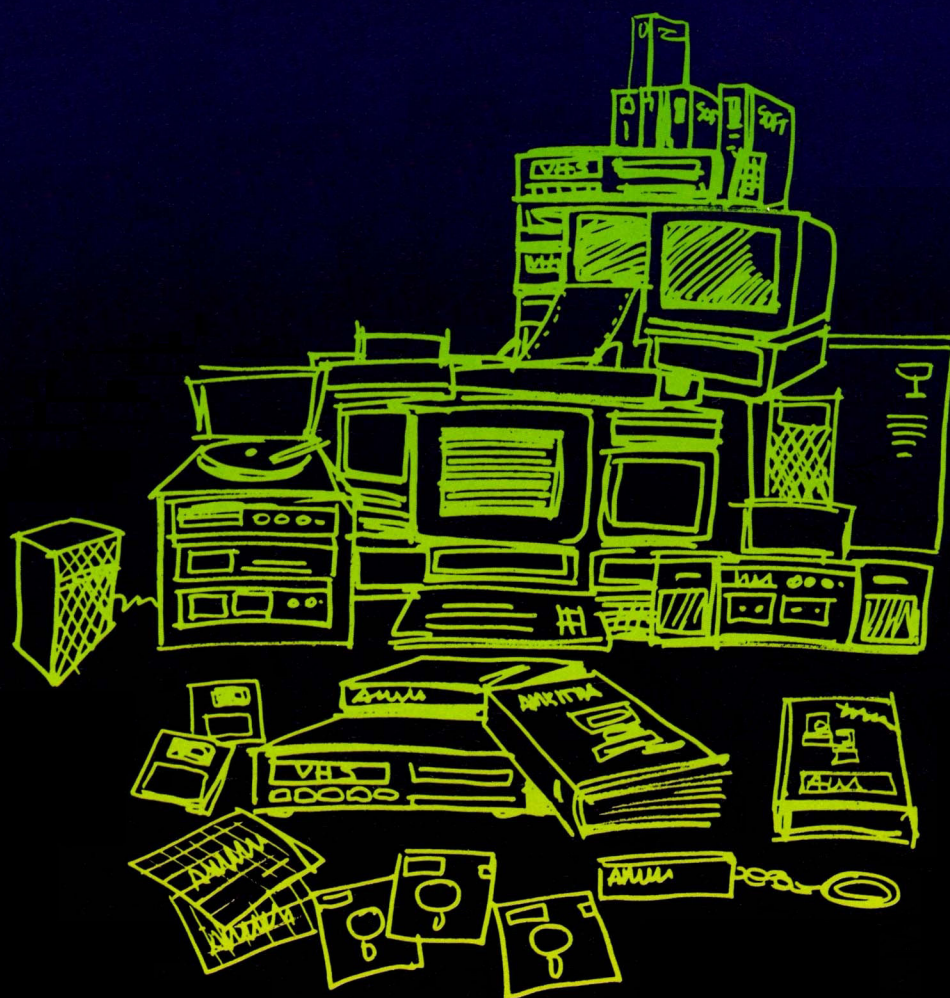
Notas:
— Examinem tudo o que acharem possível.
— Usem o spray do hálito, quando as mensagens de mau hálito surgirem com muita frequência.
— Cuidado com os becos escuros (Larry não é bom a dar socos).
— Nunca entrar no táxi com uma garrafa de vinho ou outra bebida (o condutor adora beber e esquece-se que vai a conduzir).
— Usar o preservativo com a prostituta (linguagem para maiores de 16 anos) — (desculpem) pois ela tem uma doença venérea.
— Apanhem tudo o que for possível (menos coisas desagradáveis...).
Deixamos ainda algumas coisas "no ar" para os nossos colegas resolverem, tais como:
O convencer o namorado de Faith, o usar a revista, o relógio, o telefonar para a empresa produtora do jogo, etc.
No entanto as soluções são óbvias.
Com esperança de vermos o nosso trabalho publicado, despedimo-nos desejando votos de prosperidade para a vossa revista.
Com simpatia e sinceros agradecimentos:

Carlos Jorge Moutinho de Pina Neves e Ricardo Manuel Pedrosa Pereira

DESTACÁVEL

clube

AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

PROGRAMAS DISPONÍVEIS

VER DESCRIÇÃO NOS NÚMEROS ANTERIORES
DA AMSTRAD MAGAZINE

- FS-101 BUGS
- FS-102 PINBALL
- FS-103 PITFALL
- FS-104 POKER MACHINE
- FS-105 PYRAMID
- FS-106 RAIN
- FS-107 ROCKETS
- FS-108 XWING
- FS-109 MAHJONG
- FS-110 MATH PAK
- FS-111 EPISTAT
- FS-112 MAHJONG — para ecrã EGA
- FS-113 ALLMAC
- FS-114 ICON MAKER
- FS-115 ALTAMIRA — editor gráfico
- FS-116 DRAW POKER
- FS-117 PIANO MAN
- FS-118 UTILITÁRIOS PARA ECRÃS EGA
- FS-119 WORLD
- FS-120 MUSIC
- FS-121 PAINT
- FS-122 FXMATRIX
- FS-123 BIORRITMO VERSÃO 3.0
- FS-124 TAROT
- FS-125 BLACK JACK
- FS-126 GIN RUMMY
- FS-127 EDWIN
- FS-128 MONOPOLY
- FS-129 ANSIDRAW
- FS-130 CASIOZ
- FS-131 BIORRITMO PESSOAL
- FS-132 BACCARAT
- FS-133 I`CHING
- FS-134 ANSI-ANIMATOR
- FS-135 MAIL
- FS-136 LABEL
- FS-137 TEMAS MUSICAIS
- FS-138 TWCALC22
- FS-139 ORIGAMI
- FS-140 GAMÃO
- FS-141 PRODIAGS
- FS-142 EMULADOR DE Z80 E CP/M 2.2
- FS-143 SPOOLER P/ MS-DOS
- FS-144 EMULADOR DE CGA PARA CARTA GRÁFICA HERCULES
- FD-901 STAR-SAK, PC-SIZE, FORGET-IT, PC-PLAN, PC-EMS, PC-MULTI, PC-PITMAN
- FD-902 TRIVIA MACHINE
- FD-903 UTILITÁRIOS PARA O WORDSTAR

GEM GRAPH + GEM DRAW

GEM GRAPH — Com a simples movimentação do rato e premindo apenas um botão, podemos obter gráficos profissionais de alta qualidade: de barras, tipo tarte com ou sem explosão, de símbolos, de linhas ou de mapas. Do tamanho e estilo que você decidir; com texto, cores e fundos de relevo para dar ao seu gráfico um aspecto tridimensional.
Gem Graph é um programa com excelentes qualidades gráficas.

GEM DRAW — Desenhos lineares, artísticos, organigramas, esquemas, etc. Escolha os elementos no menú e dê largas à sua imaginação. GEM DRAW converterá o seu PC num estúdio profissional com 6 tamanhos e tipos de letra, 20 livrarias de gráficos disponíveis, 39 funções de trama, régua, alinhamento, etc. e quando o seu desenho estiver perfeito, obtenha a cópia impressa em papel ou transparência.

PREÇO: 24 900$00 REF. 302, postal 3

SIDEKICK

STOCK LIMITADO

Software concebido para estar instalado no seu computador, em cima da sua secretária para:

- Cálculos rápidos
- Bloco-notas
- Editor de textos compatível WordStar/Turbo Pascal
- Agenda telefónica
- Planeamento de actividades
- Ligação automática de chamadas telefónicas
- Registo de recados e mensagens
- Pesquisa de códigos ASCII

Carregue de manhã o SIDEKICK na memória do computador e fique acompanhado durante todo o dia com esta poderosa ferramenta de trabalho, mesmo utilizando o computador para explorar outro software.

PREÇO: 3 900$00 REF. 303, postal 3

MANUAL DE BASIC 2 PARA PC

STOCK LIMITADO

Ainda não sabe BASIC? Já conhece outro BASIC? Mas não conhece o BASIC 2!
Esta é a linguagem de programação que lhe faz falta conhecer. As sua potencialidades são muitas e convidamo-lo a vir descobri-las.
Através da utilização das janelas do GEM você estabelece um diálogo permanente com a máquina.
O BASIC 2 utiliza, para além de muitas outras particularidades que não encontram nas versões de BASIC disponíveis no mercado, ficheiros indexados próprios das linguagens de gestão. Esta é uma das muitas características que o distingue dos outros. E, concerteza, muito mais.
Este é o manual que lhe faz falta na sua secretária. Não perca a oportunidade de adquirir o manual ao preço... bem... ao preço AMSTRAD.

PREÇO: 2 690$00
REF. 304, postal 3

PC

STOCK LIMITADO

SUMMER GAMES

O quê ?!! O Verão já passou e você não está minimamente interessado no "Summer Games" ?

Está bem, não adquira este jogo e mantenha-se com essa má forma habitual até chegar o próximo Verão. Ao fim e ao cabo nem todos podem andar na praia a mostrar um corpo "Tarzanico".

PREÇO: 1 900$00 REF. 327, postal 3

Coberturas para impressora AMSTRAD DMP 3000 e DMP 3160

PREÇO: Elicalfe 2 000$00 REF. 202, postal 3

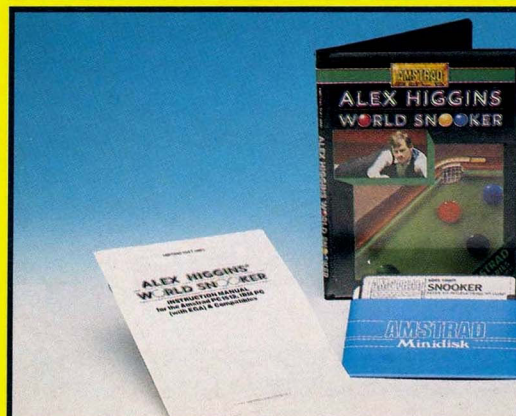
WINTER GAMES

PC

STOCK LIMITADO

Não diga que nunca viu neve, ou que só viu gelo dentro de um copo de martini, com este conjunto de jogos de Inverno compilados numa única disquete de 5.25" veja neve, veja gelo, e... se possivel, vá "vendo" uns "martinizitos" enquanto joga.

PREÇO: 1 900$00 REF. 328, postal 3

PC

STOCK LIMITADO

ALEX HIGGINS SNOOKER

Jogar snooker sempre foi aquilo que a vida lhe deu de melhor. Aliás, uma mesa de snooker no mesmo local onde hoje possui a mesa da sala de jantar, talvez até nem lhe pareça uma ideia assim tão descabida. No fundo, uma mesa é uma mesa, e só é pena que nem todas lhe permitam jogar snooker. Mas... e se em vez de encher a casa com o "hardware" necessário para jogar snooker, você jogasse snooker no ecrã do seu PC Amstrad, acompanhado por ALEX HIGGINS? Pelo menos sempre podia continuar a jantar numa mesa sem buracos e evitava tropeçar nos bocados de giz espalhados pelo chão.

PREÇO: 1 900$00 REF. 329, postal 3

MICROSOFT FLIGHT SIMULATOR version 3.0

Suporta todas as cartas gráficas desde CGA a VGA

Para quem gosta de simuladores de vôo este é **O SIMULADOR DE VÔO**.

Suportando muitas das cartas gráficas habituais nos PC´s inclusivé a Hercules, a EGA, a VGA, e a CGA em visores de cristal liquido ou CRT´s, o Flight Simulator que neste número colocamos à disposição de todos os leitores foi concebido por uma das maiores softhouses da actualidade, senão mesmo a maior - a Microsoft - e é no minimo um simulador excelente a todos os níveis. Em termos de gráficos, por exemplo, para além de suportar as cartas gáficas já referidas e de delas extraír as capacidades que lhes são inactas, suporta ainda outras cartas gráficas não previstas na versão base mas adicionaveis através de drivers externos.

A simulação que pode decorrer num de três aviões diferentes, escolhido pelo utilizador, pode basear-se em operações de descolagem, aterragem, ou vôo normal, sofrendo, ou não, efeitos climatéricos (chuva, vento, neve, etc), ou temporais (dia, fim de tarde, noite, etc.), e estando, ou não, condicionada a um conjunto enorme de outros factores, entre os quais podemos referir os vôos em esquadrilha, ou em perseguição, quer em periodos de paz, quer em periodos de guerra.

O nível de realidade da simulação e controlavel pelo utilizador através de opção acedida por teclado, e para os utilizadores menos à vontade num "cock pit" existe ainda a possibilidade de assistir a lições de vôo sub-divididas por tarefas a executar. A documentação é composta por um enorme manual, diversos mapas, e um pequeno livro de "Quick Reference" (referências rápidas), apoiando de uma forma melhor do que excelente o jogo que se encontra dividido pelas duas disquetes de 5.25" que complementam a package.

Para além do interesse do jogo, pensamos que é digno de nota o facto dele suportar e tirar proveito das cartas VGA, facto que, sem dúvida, o torna único no mercado português.

PREÇO: 9 900$00 — REF. 330, postal 3

QUICK BASIC versão 3.0

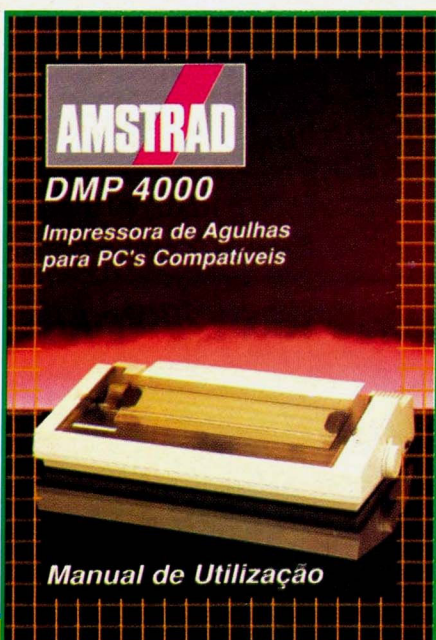
SUPORTA O PROCESSADOR ARITMÉTICO 8087

Uma excelente linguagem de programação e um óptimo compilador de programas concebidos em BASICA ou GW-BASIC, o Quick BASIC proporciona a todos os programadores desta linguagem uma velocidade de processamento que embora não sendo tão grande como a que se obtém no dialecto da mesma linguagem lançado pela Borland, é muito mais standard.

Para todos os utilizadores do GW, o Quick BASIC só pode ser a evolução perfeita. Baseado num set de instruções que quase se pode considerar cem por cento igual ao do dialecto GW, o QB traz-nos toda a velocidade de uma linguagem compilada, as facilidades de "debugging" comuns aos interpretadores da mesma linguagem, e um completo manual de utilização, por um preço impossivelmente baixo!!!

PREÇO: 15 000$00 — REF. 331, postal 3

DMP 4000

STOCK LIMITADO

- MANUAL DE UTILIZAÇÃO EM PORTUGUÊS

Com uma qualidade de impressão relativamente elevada tendo em consideração que se trata de uma impressora de 9 agulhas, a DMP 4000 pode distinguir-se actualmente como uma impressora bem sucedida no mercado nacional. Tal facto, constituiu uma das razões que nos levou a optar pela inclusão do seu manual de utilização, EM PORTUGUÊS, nesta secção da AM, procurando com isso continuar a proporcionar aos nossos leitores informação tão detalhada quanto possivel, numa linguagem tão simples quanto possivel, a um preço nitidamente impossivel.

PREÇO: 500$00 — REF. 320, postal 3

LOCOSCRIPT 2 (para PCW 9512) — Manual do Utilizador EM PORTUGUÊS

STOCK LIMITADO

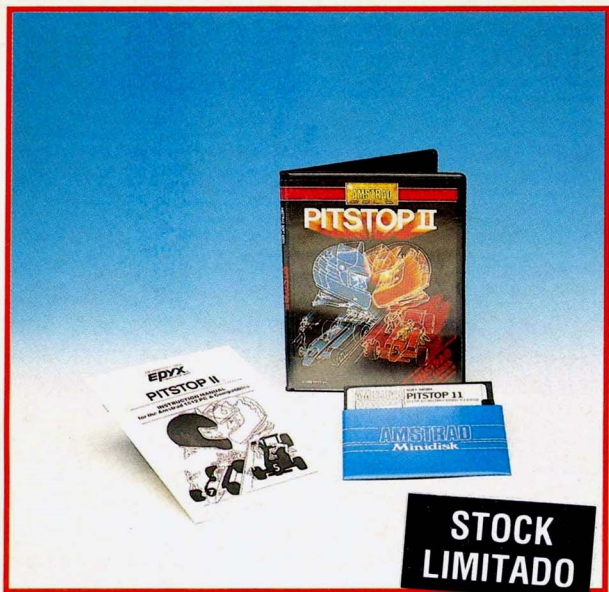
Quase quatrocentas páginas de texto, figuras, esquemas, e exemplos, constituem o mais completo livro em português sobre um processador de texto que tem arrastado centenas de pessoas dos teclados das máquinas de escrever para os teclados das modernas máquinas de processamento de texto.

Dividido em quatro partes distintas o manual do Locoscript que aqui apresentamos inicia o seu passeio pelo processador de texto em causa, através de uma passagem pelas "Noções Básicas" e "Refinamentos", concluindo a dissecação do tema com as "Funções Avançadas" disponiveis, e complementando todas estas partes e informações com um detalhado, e bem estruturado, Apêndice, repartido por 5 assuntos diferentes. Tudo o que o utilizador do Locoscript 2 precisa saber para resolver eventuais problemas menos comuns, ou apenas escrever uma simples carta, pode encontrar-se neste manual ao cabo de meia dúzia de segundos de procura.

Utilizando um conhecido slogan há algum tempo passado no pequeno écran podemos mesmo dizer que: se já possui um PCW, utiliza o Locoscript 2 e não possui este manual, DE QUE É QUE ESTÁ À ESPERA ?!!

PREÇO: 1 200$00 — REF. 322, postal 3

Coberturas para computador AMSTRAD PC 1512 e PC 1640

PREÇO: Elicalfe 4 530$00 — REF. 201, postal 3

PITSTOP II

As corridas de carros nunca foram um desporto para praticar sozinho. PITSTOP II, vem precisamente comprovar essa afirmação ao permitir que dois jogadores possam correr simultâneamente, lutando pelo primeiro lugar. Se, no entanto, não tiver alguém contra quem competir, o seu computador pessoal estará sempre pronto para lhe fazer frente numa dura corrida ao longo de qualquer uma das três pistas possiveis.

O objectivo principal é simples: ganhar a corrida. Contudo para conseguir atingi-lo você terá que tomar várias decisões estratégicas enquanto garante que os seus adversários não o ultrapassam. Os pneus, por exemplo, ou a falta de combustivel, podem obrigá-lo a parar, resta-lhe a si a opção de efectuar as paragens necessárias ou de se arriscar a ficar a meio do caminho. No fundo como, já referimos, entre o primeiro e o último lugar existe apenas uma enorme dose de perícia, um pouco de bom-senso, e uma boa estratégia.

PREÇO: 1 900$00 REF. 323, postal 3

MEAN 18

Gostava de praticar golf mas não tem espaço no local onde vive, e como alternativa, tentar jogar com os buracos que existem na estrada por onde passa diariamente, parece não ser a melhor solução.

Se é esse o seu problema, não se preocupe nem mais um minuto com ele. O MEAN 18 transforma o seu PC num óptimo campo de golf, e deixa-lhe a si a possibilidade de executar magnificas tacadas, ou até mesmo de modelar o campo de acordo com o seu gosto pessoal, evitando a cansativa tarefa de carregar os tacos, ou andar várias centenas de metros atrás da pequena bola branca, que parece bater em todo o lado menos no fundo dos buracos.

Jogar golf pode ser muito mais fácil!

PREÇO: 1 900$00 REF. 324, postal 3

THE AMSTRAD COLLECTION

Quatro jogos, três disquetes, dois minutos a preencher o postal para os encomendar, uma única oportunidade de adquirir tudo isto por este preço.

Jogos incluidos nesta package:

THE DAM BUSTERS - SYDNEY
BRUCE LEE - DATASOFT
PSI-5 TRADING COMPANY - ACCOLADE
TAG-TEAM WRESTLING - DATA EAST

PREÇO: 1 900$00 REF. 321, postal 3

MICROSOFT WORKS

Descrever o WORKS em tão pouco espaço, seria completamente impossivel, para além de que estariamos apenas a repetir aquilo que a maior parte dos utilizadores já ouviu acerca desta package integrada. No fundo em tão poucas linhas apenas podemos dizer que o WORKS integra quatro poderosas ferramentas prontas para satisfazer a maior parte das necessidades informáticas de qualquer utilizador.

Processador de texto, folha de calculo, e base de dados, são apenas 3 das 4 aplicações integradas nesta package. A quarta aplicação pode funcionar como complemento de cada uma destas ou independente de todas elas, visto que se trata de uma package de comunicações.

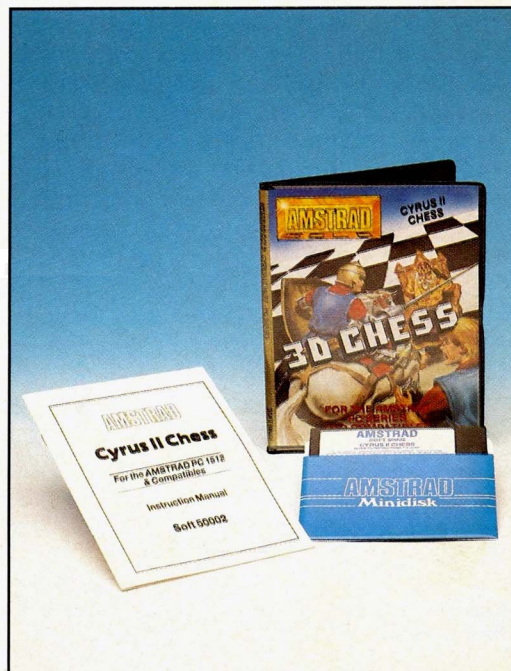
A complementar as 12 disquetes fornecidas (8 disquetes em formato 5.25", e 4 com o mesmo conteúdo em formato 3.5") um extenso e completo manual com mais de 600 páginas ordenadas de uma forma lógica, e incluindo um completo, e útil, indice, torna o WORKS a package ideal para quem tem pouco tempo para aprender a "mexer" no computador mas deseja aproveitar todas as suas potencialidades.

"Um dia de trabalho numa hora de WORKS", podemos afirmar que é a melhor forma de descrever o que esta "pequena maravilha" pode fazer por si. Tudo o resto está dito nas entrelinhas do que dissemos, e demonstrado no software que lhes deu origem.

PREÇO: 37 500$00 REF. 325, postal 3

CYRUS II CHESS

Adquirir o Cyrus II é adquirir um excelente jogo de xadrês com as seguintes caracteristicas:

* tabuleiro a duas ou três dimensões, comutaveis a qualquer momento através do teclado;
* suporte de carta gráfica CGA e EGA 16 cores (PC 1640 ECD);
* 16 níveis de dificuldade;
* possibilidade de executar cópias da partida em papel, através de qualquer impressora do tipo da DMP 3000, DMP 4000, ou compatíveis;
* possibilidade de gravar o jogo no meio de uma partida para posteriormente o retomar no mesmo ponto;
* inúmeras opções de análise e ajuda durante a partida;
* um completo manual de 27 páginas;
* um preço de fazer rir.

PREÇO: 1 900$00 REF. 326, postal 3

TODOS OS PREÇOS INCLUEM O TRANSPORTE E O I.V.A. A 17%

CM1 — CONJUNTO DE 5 JOGOS SORTIDOS PARA CPC

STOCK LIMITADO

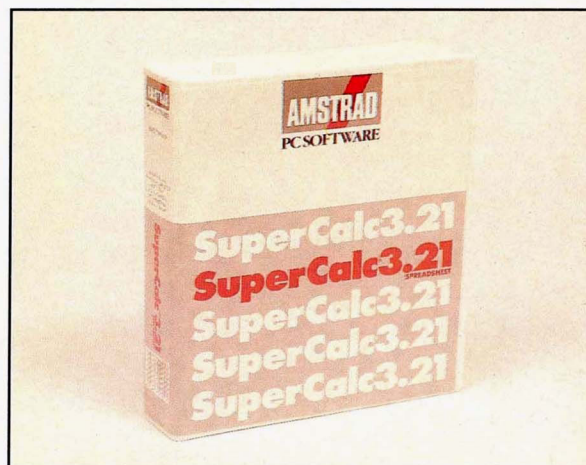
Se é possuidor de um CPC, se tem entre 5 e 95 anos, se tem tempo para jogar e não tem jogos — então tem um grave problema.
Felizmente nós propomos-lhe uma solução.
5 Cassetes com 5 jogos (surpresa) diferentes, vão diverti-lo por muito mais de 5 horas e custar muito menos de 5 contos, embora também custem um pouco mais de 5 escudos.

PREÇO: 990$00 REF.313, postal 3

TurboCAD

De instalação fácil, e utilização simplificada como consequência do funcionamento baseado em menus tipo "pop-up" o TurboCAD pode ser o utilitário que você procura para "dar asas á sua imaginação" no dominio do desenho técnico.
Acompanhado por um completo manual que lhe permite entrar sem grandes dificuldades no mundo do Desenho Assistido por Computador, o TurboCAD assegura a compatibilidade com o AutoCAD (uma das "packages" de CAD mais populares entre os utilizadores de computadores), sendo cerca de 9 ou 10 vezes mais económica do que esta última.

PREÇO: 27 500$00 REF.318, postal 3

SUPERCALC 3.21

O standard em folhas de cálculo é, ainda hoje, nitidamente imposto pelo LOTUS 1,2,3. Ninguém sequer coloca isso em causa. O que começa a colocar-se em causa são as vantagens de utilização desta folha de cálculo numa altura em que existem dezenas de outros utilitários com o mesmo fim, compativeis com o LOTUS, mas... muito mais possantes.
É este, por exemplo, o caso do SuperCalc, agora disponivel na sua versão 3.21.
O SuperCalc foi uma das "packages" que soube tirar proveito do facto de não "rasgar" mercado.
Aproveitando os resultados das experiências dos seus "adversários", o SuperCalc 3.21 melhorou muitas das suas caracteristicas, apresentando por exemplo, entre muitas outras qualidades dignas de nota, modos de representação gráfica superiores aos que a maior parte dos utilitários deste tipo incluem, uma boa velocidade de processamento de dados, e um conjunto de "HELP screens" mais do que suficiente para se começar a tirar proveito da "package", mesmo antes de se começar a ler o detalhado manual que a acompanha.
Conclusão: se nunca utilizou uma folha de cálculo, o SuperCalc 3.21 é-lhe indispensável; se já utiliza uma folha de cálculo o SuperCalc 3.21 é-lhe indispensável.

PREÇO: 19 900$00 REF.319, postal 3

TODOS OS PREÇOS INCLUEM O TRANSPORTE E O I.V.A. A 17%

PC

EXCLUSIVO DO CLUBE DE LEITORES

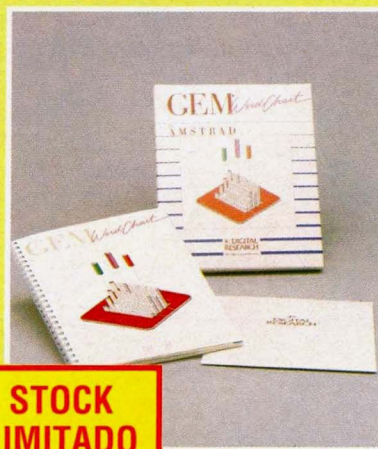
JÁ NÃO PRECISA DE SAíR DE CASA PARA IR JOGAR

POKER

AO CASINO

O jogo Good Luck é uma réplica do popular Poker das máquinas dos casinos, permitindo todo o tipo de jogadas — 2 pares, sequência, fullen, etc. e, para os mais destemidos, dobrar ou perder

PREÇO: 2 000$00 REF.306, postal 3

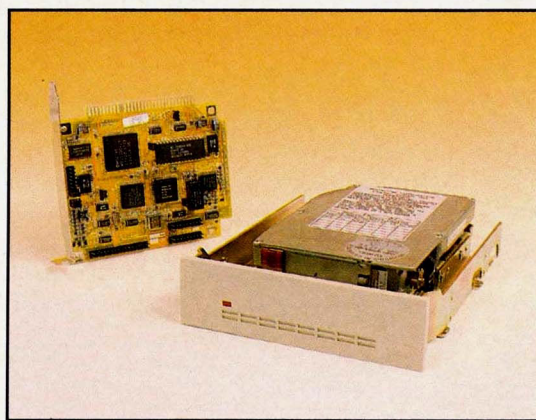
GEM WORDCHART

STOCK LIMITADO

Actualmente, mais de 80% das apresentações são feitas através de palavras — e não de gráficos.
O GEM WORDCHART, concebido com a intenção de lhe servir de instrumento de trabalho na realização simples de apresentações, permite a utilização de diversos tipos de letras com recurso a inúmeras variantes de cada tipo, selecção de limitadores e formatos, e combinação de cores, através de menus do tipo "drop-down".
Para lhe tornar a composição da folha mais fácil, o texto aparece no écran exactamente igual à posterior cópia impressa, e a largura das colunas pode seleccionar-se com a simples pressão de um botão do "mouse".
Em resumo, o GEM WORDCHART, situa-se entre o PRINT MASTER e o PAGE MAKER, apresentando no entanto, em relação a um e a outro, algumas vantagens na concretização de pequenos trabalhos.

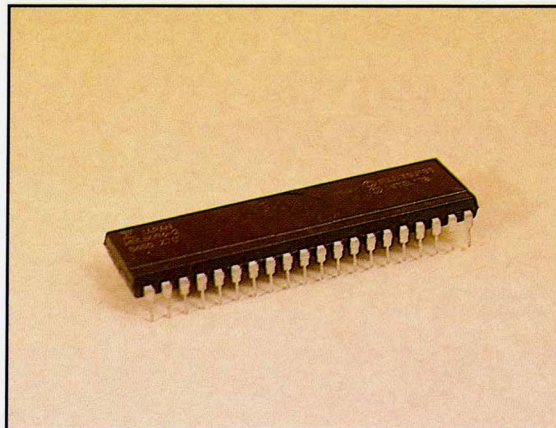
PREÇO: 9 900$00 REF. 308, postal 3

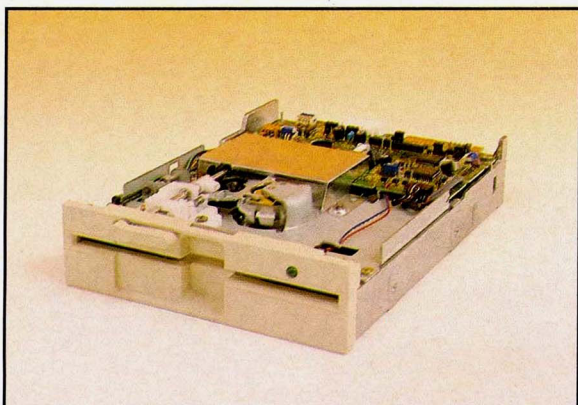
DISCO RIGIDO DE 30 MB

COM VENTOÍNHA E CONTROLADOR

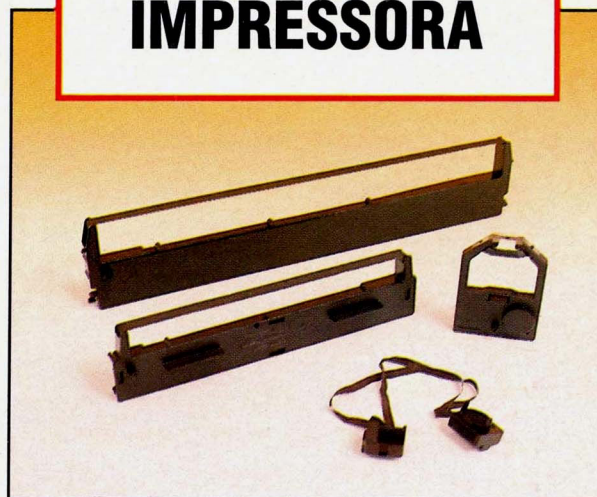
Resultado de uma deficiente análise das necessidades pessoais, de um investimento em meios informáticos necessariamente baixo, ou de qualquer outra razão menos generalizada, a aquisição de computadores pessoais com uma ou duas drives sempre foi superior à de equipamentos com uma memória de massa de maior capacidade. Consequência desse facto, é quase sempre a posterior troca do equipamento adquirido, ou a incessante procura de um disco rigido com uma capacidade de armazenamento razoável, e um custo "impossivelmente" baixo.
O disco que lhe propomos pode não ser aquele que melhor lhe convem em termos de preço, mas é concerteza uma boa aquisição, se o relacionarmos com os restantes componentes deste tipo já existentes no mercado nacional. De qualidade excelente (diga-se a propósito que o controlador que o acompanha é da Western Digital), este disco será sempre a sua melhor aquisiçãol, se o seu computador pessoal ainda não inclui um semelhante. Ofreça-o a si mesmo, você merece.

PREÇO: 119 000$00 REF. 901, postal 3

PROCESSADOR ARITMÉTICO INTEL 8087 (8MHZ)

Se lhe dissessem que o seu computador pessoal em determinadas situações pode funcionar com uma velocidade de processamento cem por cento superior áquela em que neste momento funciona, estariam sem dúvida a pensar na simples inserção de um processador aritmético na placa principal do seu PC. Tarefa que mesmo uma criança poderá levar a cabo com sucesso a inserção do circuito integrado INTEL 8087 no suporte a ele destinado na placa principal dos PC´s Amstrad, pode com efeito, em certas situações, duplicar a velocidade de processamento da máquina em que está inserido, aumentando-a sempre consideravelmente em todas os outros casos.
Imagine, por exemplo, a velocidade que a sua aplicação em Turbo BASIC, Turbo Pascal, ou Turbo C (para não citar muitas outras) pode atingir com a adição de um simples integrado ao hardware já disponivel, isto para não falar das aplicações de CAD que costuma utilizar, ou de todas as outras aplicações "pesadas" que entretanto recusou por "trabalharem a vapor" numa máquina da era nuclear.

PREÇO: 54 000$00 REF. 902, postal 3

DRIVES DE 5.25"

Em tempos adquiriu um PC com uma única drive, e agora deseja adicionar-lhe uma segunda drive de 5.25" esta oferta soluciona-lhe o problema. Fácil de instalar com alguma habilidade, e uma dose igual de paciência e tempo livre, esta drive de 5.25" vai poupar-lhe o dinheiro que o técnico lhe leva para proceder a uma instalação deste tipo, proporcionando-lhe muito mais gozo pessoal por no final da operação poder afirmar que foi você quem fez a instalação da drive.
Concluida a instalação, você ganhou mais experiência, e... sobretudo ganhou mais dinheiro.

PREÇO: 15 000$00 REF. 903, postal 3

FITAS PARA IMPRESSORA

Por muito boa que seja uma impressora, mais tarde ou mais cedo ela acaba sempre por nos aborrecer. Talvez no futuro as impressoras consigam produzir automáticamente os seus próprios consumiveis, mas por agora somos nós que penosamente os temos de adquirir. As fitas para impressora, nitidamente inseridas nesta categoria, possuem um preço cada vez mais elevado e, apesar disso, são muitas vezes dificeis de encontrar na loja onde costumamos fazer as nossas compras informáticas. Por esta razão nada melhor do que comprar as fitas de que necessita, quando necessita, sem sequer ter de se preocupar em encontrá-las, ou mesmo ter de se deslocar para adquiri-las.

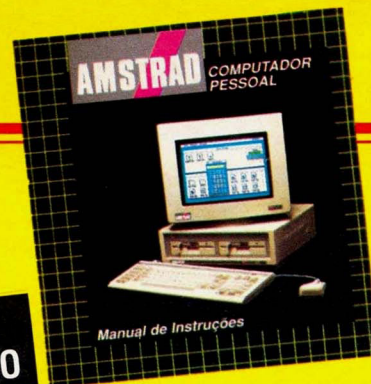
DMP 2/3/3160	1 400$00	REF. 904	postal 3
DMP 4000	2 100$00	REF. 905	
LQ 3500/PCW	1 400$00	REF. 906	
LQ 5000	2 650$00	REF. 907	

STOCK LIMITADO

DISKETTES AMSTRAD

Em 3", 3.5", ou 5.25" as diskettes Amstrad são fornecidas em conjuntos de 10 unidades com caixa plástica, garantindo uma perfeita formatação e fiabilidade dos dados armazenados.

3"	PREÇO: 8 490$00	REF. 315	postal 3
3.5"	PREÇO: 5 990$00	REF. 316	postal 3
5.25"	PREÇO: 2 690$00	REF. 317	postal 3

STOCK LIMITADO

MANUAL DO PC EM PORTUGUÊS

Será que os computadores só podem ser utilizados por quem sabe inglês?
É evidente que não. Embora o conhecimento da língua inglesa facilite a aprendizagem, nunca se poderá considerar indispensável para este efeito. No nosso país, são cada vez mais frequentes as marcas que traduzem os manuais e as packages, e adaptam os teclados, para poderem possuir boas soluções informáticas em mercados que nada têm a ver com a língua inglesa.
Foi assim, seguindo esse princípio, que AM optou por incluir nesta secção a tradução do MANUAL DO PC, para facilitar a vida a todos os que em Portugal preferem ler em português.

PREÇO: 1 900$00 REF. 310, postal 3

FORTH P/ CPC

Num momento em que começam a surgir no mercado alguns processadores que possuem como linguagem "natural" o FORTH, torna-se interessante poder oferecer aos possuidores dos CPC a hipótese de experimentar o poder desta linguagem como forma de comunicar com a máquina. Com algumas vantagens sobre o BASIC (nomeadamente uma maior velocidade de processamento), o FORTH continua a manter inúmeros adeptos entre os programadores e utilizadores de computadores, que não hesitam em defendê-lo, em muitas situações, como uma das melhores linguagens de programação.

APRESENTADO EM CASSETE

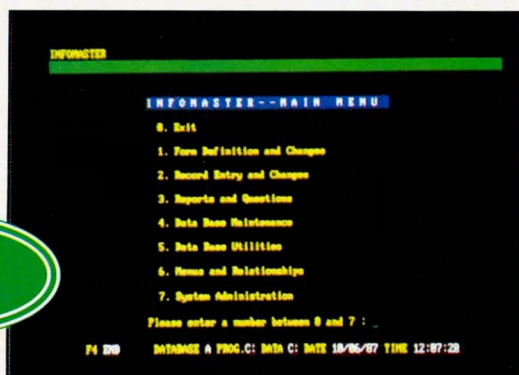
PREÇO: 900$00 REF.314, postal 3

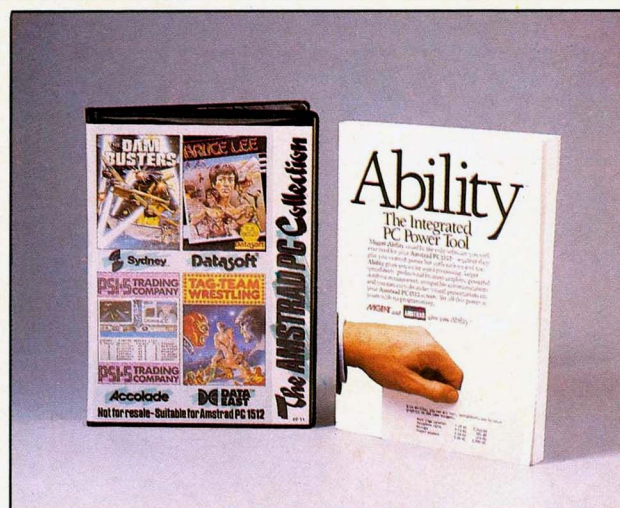
MANUAIS TÉCNICOS DO PC 1512 E PC 1640

Para a maior parte dos utilizadores dos computadores pessoais Amstrad, a informação contida na documentação que acompanha o PC, é mais do que suficiente para conseguir tirar dele todo o proveito que sempre se visou como objectivo desde o momento da sua compra. Para alguns outros, no entanto, as necessidades são um pouco diferentes e para melhor poderem manipular a máquina com que habitualmente trabalham, um pouco mais de informação é, no mínimo, desejável.
Os manuais técnicos do PC 1512 e PC 1640 incluem mais informação sobre a sua máquina do que aquela que provavelmente irá necessitar. Vão, portanto, além da referida "informação desejável". No fundo, como a própria designação deixa entender estamos a falar de manuais técnicos, bastante técnicos, para técnicos, ou futuros... técnicos.

PC 1512 PREÇO: 7 700$00 REF. 908, postal 3
PC 1640 PREÇO: 7 700$00 REF. 909, postal 3

INFOMASTER

PC

Em bases de dados, é verdade que o DBASE criou um standard, mas não é menos verdade que INFOMASTER ultrapassou esse standard.
Permitindo uma utilização eficiente após alguns minutos de trabalho, possibilitando a utilização de um máximo de 65535 registos em cada ficheiro, e um máximo de 255 campos em cada registo, o INFOMASTER torna-se o sistema de gestão de base de dados mais adequado para as pequenas empresas.
Funcionando num sistema de menus que permite a fácil manipulação de informação, e a configuração da base de dados por utilizadores com um mínimo de conhecimentos, esta package utiliza parte da RAM como cache, conseguindo deste modo uma velocidade que em determinadas situações se pode considerar cerca de 400% superior à das bases de dados convencionais.

PREÇO: 17 900$00 REF. 311, postal 3

ABILITY + 4 JOGOS

Package integrao de programas que lhe oferece:

PC

a) Base de Dados.
b) Folha de Cálculo.
c) Gráficos de Gestão.
d) Processamento de Texto.
e) Comunicações.
f) Gerador de Apresentações.

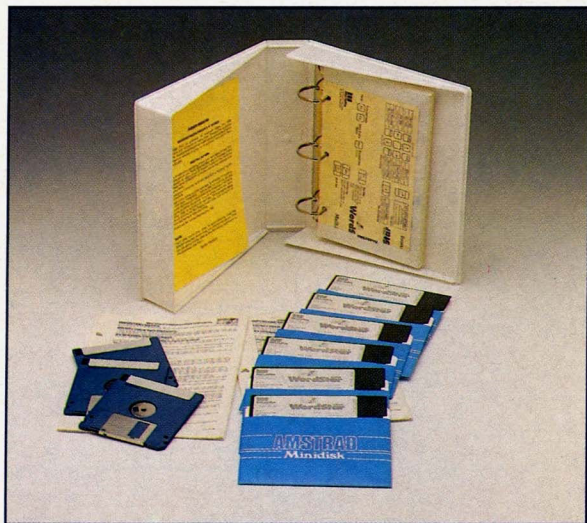
Incluindo:

1) Manual de fácil leitura e manuseamento.
2) Utilização compartilhada de dados para as diferentes aplicações.
3) Integração activa entre os programas, (não realizável em programas conhecidos do mercado).
4) Com o programa APRESENTAÇÃO, incluído no Ability, podem preparar-se informações obtidas com os dados manuseados com o programa base.

E ainda 4 Jogos: "The Dam Busters", "Bruce Lee", "Psi 5 Trading Company" e "Tag Team Wrestling".

PREÇO: 8 900$00 REF.301, postal 3

CLUBE AM

WORDSTAR EXPRESS

Apesar dos inúmeros processadores de texto e programas de DeskTop Publishing, que têm surgido no mercado de software durante os últimos anos, o WordStar continua a manter um número considerável de adeptos, não só porque sendo uma processador de texto pouco "pesado" consegue manter um conjunto de caracteristicas muito interessante e comum a aplicações mais complexas, mas também porque permite ao utilizador o processamento de texto num formato que cada vez mais se vai implantando como um standard.
O WordStar Express, seguindo a linha das anteriores versões deste programa é, "apenas", um standard com melhoramentos.

PREÇO: 19 900$00 REF. 312, postal 3

DISKETTES DE 3.5" E 5.25"

Para quem não faz questão em utilizar apenas disquetes Amstrad, e quem quer poupar algum dinheiro com a compra das tais caixas de disquetes que há muito tempo necessitava, decidimos tornar disponiveis 3 novas marcas de disquetes de 5.25" e uma de 3.5" a preços super-baixos.

5.25"	MEMOREX Cx. de cartão c/ 10 disq.)	PREÇO: 1 350$00	REF. 332
	MOORE (Cx. de Plást. c/ 10 disq.)	PREÇO: 2 200$00	REF. 333
	RPS (Cx. de cartão c/ 10 disq.)	PREÇO: 1 700$00	REF. 334
3.5"	RPS (Cx. de cartão c/ 10 disq.)	PREÇO: 4 000$00	REF. 335

SUPERCALC P/ CPC

Porque razão é que só os utilizadores de PC´s podem ter acesso a gestores de bases de dados, a bons processadores de texto, ou a excelentes fohas de cálculo ?
Na realidade não há razão para que o possuidor de um CPC, por exemplo, não possa aproveitar as vantagens de possuir essas mesmas aplicações.
Para começar, que tal se lhe "oferecermos" o Supercalc para CPC ?

PREÇO: 11 100$00 REF. 345, postal 3

JOYSTICKS PARA PC´s E CPC´s

Muitos dos utilizadores de PC´s em tempos tiveram o seu Spectrum, MSX, Atari, ou a sua "máquina de jogos" de qualquer outra marca. Hoje, mais ligados ás aplicações profissionais, muitos desses utilizadores pensam com saudades no seu velho computador, e nas antigas preocupações que tinham em arranjar POKE´s para o jogo do RAMBO, ou vidas infinitas para o ARKANOID.
Alguns de todos estes "tecladores de PC´s" não resistem mesmo a "experimentar" os jogos que vão aparecendo para estas máquinas, e aí meus caros, o joystick é fundamental (aceitem a opinião de um perito). Afim de facilitar a vida a todos os "experimentadores de jogos", decidimos colocar no clube AM três tipos de Joystick, de preços e caracteristicas diferentes, para que todos possam ficar satisfeitos.
Todos os joysticks apresentados funcionam sem problemas em qualquer PC 1512, ou PC 1640, desde que os jogos ou programas utilizem como controlos as teclas de cursor (esta caracteristica deve-se ao facto da entrada de joystick, presente no teclado, emular as teclas de cursor que se encontram à sua direita).
Ainda como um alerta devemos referir que o Quick Shot II é o único joystick que não funciona nos computadores da linha CPC.

Quick Shot II	PREÇO: 1 000$00	REF. 406
Quick Shot II Turbo	PREÇO: 2 750$00	REF. 407
Sheeta Star Fighter	PREÇO: 3 500$00	REF. 408

"Vale mais crédito que dinheiro"

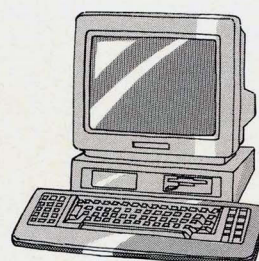
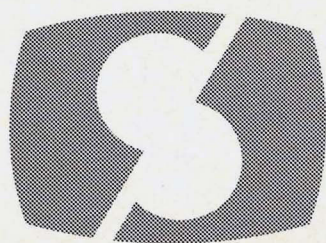
Os nossos clientes merecem-nos todo o crédito. Independentemente do dinheiro que trazem no bolso.

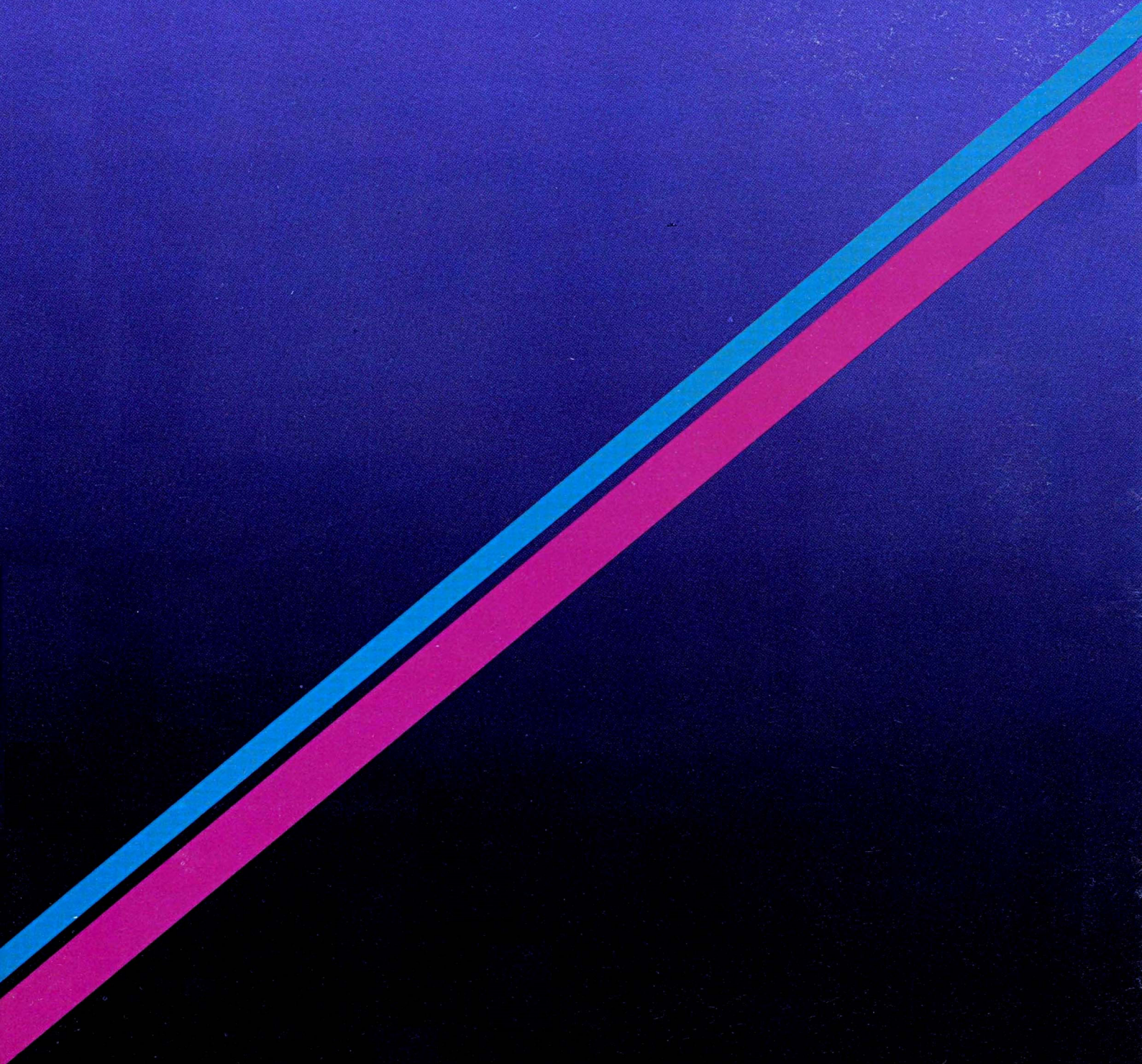
Credi AMSTRAD

Assim, oferecemos as melhores condições de crédito com amplas facilidades de pagamento — formas especiais de comercialização de onde se destacam o novo CREDI-SOCARTEL e o já conhecido CREDI-AMSTRAD.

Desta forma, quando precisar de uma boa aparelhagem de TV, Vídeo e Hi-Fi, de computadores, acessórios e outros artigos de electrónica venha ter connosco.

Pode ter a certeza de encontrar as marcas de qualidade, a assistência pós-venda garantida e um conselho profissionalizado na medida exacta das suas necessidades.

Tudo isto, englobado num novo conceito de lojas espalhadas pelo País, que aliam à variedade seleccionada o serviço impecável. Sabemos esclarecê-lo na compra do útil. E não do fútil.

LISBOA Av. Engº Duarte Pacheco, 17-19
1000 LISBOA
PORTO Rua Santa Catarina, 716
4000 PORTO
PORTO Av. da Boavista, 2881, Loja 3
4000 PORTO
GUIMARÃES R. Manuel Saraiva Brandão, 241 r/c
4807 GUIMARÃES
CHAVES Av. 5 de Outubro
Edifício Europa
5400 CHAVES
COIMBRA Av. Calouste Gulbenkian
Centro Comercial Primavera
Loja 37
3000 COIMBRA
OLHÃO R. João Rosa, 6
8700 OLHÃO
PORTIMÃO Bairro do Pontal, Bloco 2 A, c/v
8500 PORTIMÃO

SOCARTEL

A JUSTA MEDIDA DA ELECTRÓNICA

O WINDOWS NÃO TINHA... UM TECLADO PORTUGUÊS.

QUANDO apresentámos o driver para o rato Amstrad no Windows, apesar dos erros que o afectaram de inicio, muitos foram os leitores que nos falaram na hípotese de construirmos um driver para o teclado português na mesma aplicação, visto que a selecção do teclado português neste ambiente de trabalho utiliza a configuração espanhola. A todos eles nunca dissemos que não. O trabalho de concepção do driver para o teclado mostrar-se-ía, no entanto, uma tarefa relativamente morosa. Para ser mais preciso demorou exactamente um fim de semana. Mas, no final os resultados compensaram todo o trabalho, e ficámos a saber todos os segredos dos drivers de teclados para o Windows, o que nos permite, neste momento, a adaptação fácil de qualquer teclado para esta aplicação.

Assim, para além de ordenarmos todas as teclas e combinações de teclas no teclado Amstrad, aproveitámos e acrescentámos ainda dez outras combinações possiveis, onde incluimos alguns caracteres de muita utilidade. Os caracteres que decidimos acrescentar foram os seguintes: ©®$^{1}_{4}$$^{1}_{2}$$^{3}_{4}$^{0123}µ Para obter qualquer um deles basta-nos premir as teclas **Alt+Ctrl** e simultâneamente qualquer uma das dez teclas situadas na segunda fila do teclado (entre o **Q** e o **P**). Como é lógico, poderiamos ter escolhido qualquer outro conjunto de caracteres, mas uma vez analisado o set disponivel no Windows chegámos à conclusão que para além dos caracteres já disponíveis com o "acerto" do teclado, não nos fariam falta mais do que aquela escassa dezena deles que tornámos directamente acessiveis (não devemos esquecer que todos os caracteres se podem conseguir através da pressão na tecla **Alt** e da marcação do respectivo código no bloco de teclas númericas).

Expostas as razões que nos levaram a conceber este driver vejamos então a forma de o aproveitar.

Em primeiro lugar devemos começar por carregar uma versão de BASICA ou GW-BASIC e introduzir a listagem que se segue, tendo o cuidado necessário para não cometer muitos erros. Apesar do carregador utilizado verificar a validade dos dados, é possivel que alguns erros lhe escapem, e que no final o driver não funcione porque a listagem introduzida não é exactamente igual a que foi publicada (se bem que esta situação a partida seja pouco provável). Depois de introduzirmos a listagem no computador devemos gravá-la para que mais tarde a possamos rectificar (se for caso disso) sem necessitarmos de a introduzir de novo. O passo seguinte é a execução da listagem através do habitual **RUN [enter]**. Se tudo correr bem o programa dá-nos indicação disso. Se tudo correr mal, o programa também nos dá indicação disso, e neste caso devemos rectificar os erros e voltar a executá-lo.

Terminada esta fase temos o driver preparado no disco, ou disquete, que estamos a utilizar, e falta apenas copiá-lo para a disquete de SETUP do Windows, por forma a que ele substitua o ficheiro com o mesmo nome que lá se encontra.

Depois... bem, depois resta-nos instalar o Windows tal como sempre o fizemos e escolher o teclado português que, desta vez, é mesmo um teclado português.

Atenção:

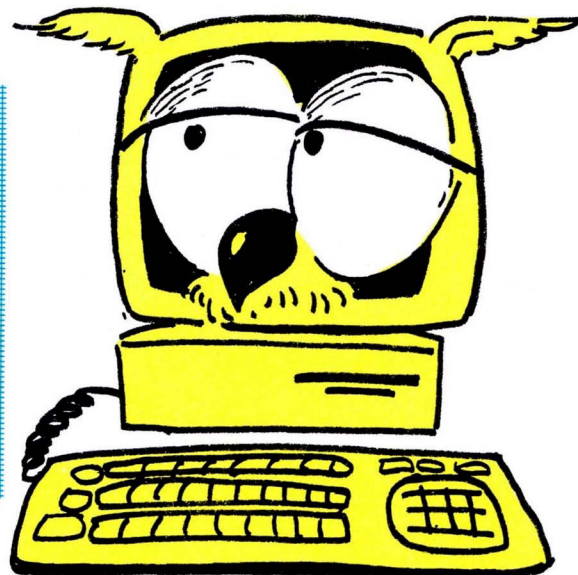
No driver do rato publicado nas AM´s nº7 e 8, cometemos um erro ao afirmar que o driver funcionaria nas versões 1.xx do Windows, visto que como posteriormente constatámos a versão 1.01 desta aplicação não o reconhece. Desta vez não vamos cometer o mesmo erro, e alertamos todos os leitores para o facto do driver em causa apenas ter sido testado no Windows 1.02, onde funciona desde há várias semanas sem qualquer problema.

Fernando Prata

```
1 REM *================================*
2 REM *                                *
3 REM *  DRIVER PARA TECLADOS AMSTRAD  *
4 REM *       UTILIZANDO O WINDOWS     *
5 REM *                                *
6 REM *     AMSTRAD MAGAZINE  1989     *
7 REM *                                *
8 REM *================================*
10 KEY OFF:CLS:SCREEN 2:TOTAL=0:LINHA=12
0:DIV=0:CORDX=10
15 PSET(8,52):DRAW"R600D14L600U14":PSET(
478,53):DRAW"D13R1U13D13E7H7"
20 LOCATE 2,1:PRINT "CRIANDO O FICHEIRO
SPAIN.DRV..........":LOCATE 8,63:PRINT"
3984 Bytes":LOCATE 12,1
30 OPEN "SPAIN.DRV" FOR OUTPUT AS #1
50 READ A$:A=VAL("&H"+A$):IF A=256 THEN
READ A$:TZ=VAL("&H"+A$):GOSUB 110
53 IF A=21845 GOTO 80
54 IF A<256 THEN TOTAL=TOTAL+A
55 IF A>256 AND A<>TOTAL THEN BEEP:PRINT
 "ERRO NA LINHA ";LINHA:PRINT "RECTIFIQU
E O ERRO DETECTADO E VOLTE A EXECUTAR ES
TE PROGRAMA.":CLOSE:LOCATE 21,1:STOP ELS
E IF A>256 THEN TOTAL=0:LINHA=LINHA+10:G
OTO 50
60 PRINT #1,CHR$(A);:GOSUB 115
70 GOTO 50
80 CLOSE
90 LOCATE 2,1:PRINT "FICHEIRO CRIADO E P
RONTO PARA SER UTILIZADO COM O WINDOWS 1
.02":LOCATE 21,1
100 END
110 A=0:A$="00":FOR M=1 TO TZ:PRINT #1,C
HR$(A);:GOSUB 115:NEXT:RETURN
115 DIV=DIV+1:IF DIV/34=INT(DIV/34) THEN
 PSET(CORDX,54):DRAW"R2D10L2U10":CORDX=C
ORDX+4
119 RETURN
120 DATA 4D,5A,8F,01,08,00,00,00,20,00,0
0,00,FF,FF,00,00,00,00,38,1D,00,00,00,00
,40,100,23,04,100,1C1,3F6
130 DATA E8,35,00,45,73,74,65,20,64,72,6
9,76,65,72,20,6E,65,63,65,73,73,69,74,61
,20,64,6F,20,57,69,6E,64,6F,77,73,20,31,
2E,30,32,20,20,0D,0A,24,0E,1F,B4,09,CD,2
1,B8,01,4C,CD,21,5A,EB,F2,100,1C4,1495
140 DATA 4E,45,04,00,5C,00,16,00,42,24,9
7,0B,05,80,02,100,C,02,00,00,00,9E,00,40
,00,50,00,50,00,5C,00,5C,00,72,04,00,00,
00,00,09,100,C,03,00,33,04,40,00,33,04,0
6,00,8F,03,41,00,91,03,08,4B,45,59,42,4F
,41,52,44,00,00,00,06,01,03,00,9D0
150 DATA 00,03,64,00,03,28,00,03,5C,02,0
3,30,02,03,46,02,00,00,5B,4B,45,59,42,4F
,41,52,44,20,41,6D,73,74,72,61,64,20,46,
2E,50,72,61,74,61,3A,33,34,21,31,A8A
160 DATA 21,33,21,32,21,31,21,30,21,41,4
D,21,50,4D,21,53,45,4B,21,2C,21,2C,21,2F
,21,3A,21,2C,21,21,33,35,31,21,31,21,33,
21,32,21,31,21,30,21,41,4D,21,50,8E4
170 DATA 4D,21,4D,4B,21,20,21,2C,21,2F,2
1,3A,21,2C,00,00,07,44,49,53,41,42,4C,45
,03,00,07,49,4E,51,55,49,52,45,01,00,06,
45,4E,41,42,4C,45,02,00,09,4F,45,8FB
180 DATA 4D,54,4F,41,4E,53,49;06,00,07,5
4,4F,41,53,43,49,49,04,00,09,41,4E,53,49
,54,4F,4F,45,4D,05,100,F1,6F5
190 DATA 8C,D8,90,45,55,8B,EC,1E,8E,D8,5
6,57,C4,7E,06,BE,00,00,B8,0C,00,8B,C8,F3
,A4,2B,C1,5F,5E,83,ED,02,8B,E5,1F,5D,4D,
CA,04,00,8C,D8,90,45,55,8B,EC,1E,162F
200 DATA 8E,D8,56,57,83,3E,20,00,00,74,1
E,B4,25,B0,09,1E,C5,16,10,00,CD,21,1F,B4
,25,B0,1B,1E,C5,16,14,00,CD,21,1F,C7,06,
20,00,00,00,5F,5E,83,ED,02,8B,E5,F78
210 DATA 1F,5D,4D,CB,8C,D8,90,45,55,8B,E
C,1E,8E,D8,56,57,C4,5E,0A,8C,C0,89,1E,0C
,00,8C,06,0E,00,83,3E,20,00,00,75,6B,B4,
35,B0,09,CD,21,89,1E,10,00,8C,06,112F
220 DATA 12,00,B8,16,35,CD,21,89,1E,1C,0
0,8C,06,1E,00,B4,25,B0,09,BA,24,00,CD,21
,C4,7E,06,E8,4E,00,B4,35,B0,1B,CD,21,89,
1E,14,00,8C,06,16,00,B4,25,B0,1B,EFB
230 DATA BA,58,01,CD,21,C7,06,20,00,FF,F
F,80,3E,9A,02,00,74,1D,C6,06,9A,02,00,1E
,B8,00,F0,8E,D8,80,3E,FE,FF,FC,1F,75,0A,
A0,C4,02,86,06,C6,02,A2,C4,02,5F,1441
240 DATA 5E,83,ED,02,8B,E5,1F,5D,4D,CA,0
8,00,B4,02,CD,16,B2,80,B4,03,BB,10,00,E8
,22,00,B4,04,B3,11,E8,1B,00,B4,08,B3,12,
E8,14,00,B2,01,B4,10,B3,91,E8,0B,12DC
250 DATA 00,B4,20,B3,90,E8,04,00,B4,40,B
3,14,84,C4,74,03,26,88,11,C3,20,AD,63,9C
,AF,79,7C,15,22,63,61,3C,AA,2D,72,5F,F8,
F1,FD,AE,27,E6,14,FA,2C,31,6F,3E,1507
260 DATA 5F,5F,5F,A8,41,41,41,AA,8E,8F,9
2,80,45,90,45,45,49,49,49,49,44,A5,4F,4F
,4F,A7,99,B0,9D,55,55,55,9A,59,5F,E1,85,
A0,83,A9,84,86,91,87,8A,82,88,89,1609
270 DATA 8D,A1,8C,8B,64,A4,95,A2,93,A6,9
4,B2,9B,97,A3,96,81,79,5F,98,C7,FC,E9,E2
,E4,E0,E5,E7,EA,EB,E8,EF,EE,EC,C4,C5,C9,
E6,C6,F4,F6,F2,FB,F9,FF,D6,DC,F8,247A
280 DATA A3,D8,4C,6C,E1,ED,F3,FA,F1,D1,F
5,D5,BF,E3,C3,6C,6E,A1,B3,A4,20,20,20,7C
,2B,2B,2B,2B,2B,2B,7C,2B,2B,2B,2B,2B,2B,
2B,2B,2B,2D,2B,2B,2B,2B,2B,2B,2B,13E8
290 DATA 2B,3D,2B,2B,2B,2B,2B,2B,2B,2B,2
B,2B,2B,2B,2B,20,20,20,20,20,5F,DF,5F,B6
,5F,5F,B5,5F,5F,5F,5F,5F,5F,5F,5F,5F,5F,
B1,5F,5F,5F,5F,5F,5F,B0;B7,B7,5F,101B
300 DATA 6E,B2,5F,5F,5A,8B,DC,1E,56,57,3
6,C5,77,08,36,C4,7F,04,FC,B4,FF,FF,E2,99
,8B,C2,5F,5E,1F,CA,08,00,E8,E1,FF,BB,94,
00,AC,3C,A0,72,04,2E,D7,22,E0,AA,184B
310 DATA 0A,C0,75,F2,EB,E1,E8,CB,FF,BB,1
4,01,AC,3C,80,72,04,2E,D7,22,E0,AA,0A,C0
,75,F2,EB,CB,8C,D8,90,45,55,8B,EC,1E,8E,
D8,56,57,8B,46,12,8B,D0,8B,5E,10,1962
320 DATA 0B,DB,7D,27,3C,12,75,1E,C4,7E,0
8,33,C0,86,06,90,03,26,38,25,74,10,0A,C0
,75,09,06,57,06,57,90,0E,E8,B3,FF,E9,52,
01,33,C0,E9,50,01,C4,76,0C,B5,80,11EC
330 DATA 26,84,6C,12,74,51,26,84,6C,11,7
5,3C,26,84,6C,10,75,45,F7,46,06,01,00,75
,3E,80,FB,53,73,39,80,EB,47,72,34,32,FF,
8A,BF,84,03,80,FF,FF,74,29,C4,7E,13CC
340 DATA 08,B0,0A,26,8A,1D,F6,E3,02,C7,2
6,88,05,3A,C3,75,B5,C6,06,90,03,01,EB,AE
,A2,32,03,BB,F9,02,83,C3,03,38,07,75,F9,
EB,29,A2,96,02,BB,82,01,83,C3,04,145D
350 DATA 38,07,75,F9,8A,47,03,3C,FF,74,1
5,26,84,6C,11,74,0F,0A,C0,79,08,24,7F,26
,84,6C,10,74,03,E9,9D,00,8B,47,01,26,84,
6C,10,74,02,86,C4,26,F6,44,14,01,102E
```

```
360 DATA 74,24,80,FA,41,72,09,80,FA,5A,7
7,04,86,C4,EB,16,88,16,82,03,BB,7A,03,83
,C3,02,38,17,75,F9,80,7F,01,FF,74,02,86,
C4,80,FA,BA,74,05,80,FA,DE,75,0D,160E
370 DATA A2,7B,03,C4,7E,08,AA,B8,FF,FF,E
B,7F,90,8A,26,7B,03,0A,E4,74,48,A2,76,03
,BB,30,03,83,C3,05,38,07,75,F9,80,7F,02,
FF,74,41,C6,06,7B,03,00,80,FC,60,1578
380 DATA 75,05,8A,47,01,EB,1E,80,FC,B4,7
5,05,8A,47,02,EB,14,80,FC,5E,8A,47,03,74
,0C,8A,47,04,80,3F,79,75,04,B0,FF,EB,2F,
3C,FF,75,2B,8A,07,EB,10,32,E4,3C,1407
390 DATA FF,75,21,E8,32,00,E9,CD,FE,3C,F
F,74,F6,C6,06,7B,03,00,C4,7E,08,50,8A,C4
,32,E4,AB,58,32,E4,AB,B8,02,00,EB,07,C4,
7E,08,AA,B8,01,00,5F,5E,83,ED,02,169C
400 DATA 8B,E5,1F,5D,4D,CA,0E,00,33,DB,B
0,40,B3,14,E8,16,00,B0,10,B3,91,E8,0F,00
,B0,20,B3,90,E8,08,00,B4,01,9C,FF,1E,1C,
00,C3,C4,76,0C,26,F6,00,01,BE,00,1340
410 DATA 00,8E,C6,74,06,26,08,06,17,04,C
3,F6,D0,26,20,06,17,04,C3,100,1CC,5D0
420 DATA FF,FE,FF,FE,08,00,0A,00,FF,FF,0
A,100,18,50,1E,33,C0,8E,D8,E4,60,3C,F0,7
3,3B,A8,80,75,3E,F6,06,18,04,08,75,30,80
,3E,49,04,04,10A8
430 DATA 72,29,8A,26,17,04,3C,45,75,05,F
6,C4,04,75,1C,3C,53,74,0D,3C,46,74,09,3C
,37,75,17,F6,C4,03,75,0B,8A,26,17,04,F6,
D4,F6,C4,0C,75,07,1F,58,2E,FF,2E,10A4
440 DATA 10,00,53,50,E4,61,8A,E0,0C,80,E
6,61,86,E0,E6,61,58,B4,80,22,E0,32,C4,50
,B0,20,E6,20,58,E8,CE,00,3C,54,75,22,0A,
E4,78,03,E9,A2,00,A1,0A,00,2E,A3,1531
450 DATA 1A,00,3D,00,F0,74,F1,A1,08,00,2
E,A3,18,00,5B,1F,58,2E,FF,2E,18,00,2E,80
,3E,23,00,00,74,03,E9,82,00,2E,FE,06,23,
00,FB,F6,06,17,04,04,74,1B,3C,46,DF0
460 DATA 75,17,2E,C6,06,22,00,00,8B,D8,E
8,6A,00,B0,03,2E,80,3E,22,00,00,75,51,EB
,54,33,DB,8A,D8,B0,FF,80,FB,54,73,44,2E,
8A,87,9B,02,32,FF,80,FB,47,72,38,1441
470 DATA 80,FB,53,77,33,F6,06,17,04,20,7
4,2C,F6,06,17,04,03,75,0D,8A,C7,0A,C0,75
,1F,2E,8A,87,A8,02,EB,18,50,53,B8,10,80,
B3,36,2E,FF,1E,0C,00,5B,58,2E,FF,1191
480 DATA 1E,0C,00,B8,10,00,B3,36,2E,FF,1
E,0C,00,2E,C6,06,23,00,00,5B,1F,58,CF,53
,51,52,56,57,06,55,CD,1B,5D,07,5F,5E,5A,
59,5B,C3,2E,FE,06,22,00,CF,B3,02,EA0
490 DATA 3C,2A,74,12,B3,01,3C,36,74,0C,B
3,04,3C,1D,74,06,B3,08,3C,38,75,0F,0A,E4
,79,07,F6,D3,20,1E,17,04,C3,08,1E,17,04,
C3,03,03,03,03,08,08,08,7F,09,09,BDB
500 DATA 09,FF,0D,0D,0D,0A,1B,1B,1B,1B,2
0,20,20,20,2B,0D,0D,0A,30,30,3D,FF,31,31
,21,FF,32,32,22,00,33,33,23,FF,34,34,24,
FF,35,35,25,FF,36,36,26,1E,37,37,BD7
510 DATA 2F,FF,38,38,28,FF,39,39,29,FF,4
1,61,41,01,42,62,42,02,43,63,43,03,44,64
,44,04,45,65,45,05,46,66,46,06,47,67,47,
07,48,68,48,08,49,69,49,09,4A,6A,DA7
520 DATA 4A,0A,4B,6B,4B,0B,4C,6C,4C,0C,4
D,6D,4D,0D,4E,6E,4E,0E,4F,6F,4F,0F,50,70
,50,10,51,71,51,11,52,72,52,12,53,73,53,
13,54,74,54,14,55,75,55,15,56,76,D80
530 DATA 56,16,57,77,57,17,58,78,58,18,5
9,79,59,19,5A,7A,5A,1A,60,30,30,FF,61,31
,31,FF,62,32,32,FF,63,33,33,FF,64,34,34,
FF,65,35,35,FF,66,36,36,FF,67,37,128A
540 DATA 37,FF,68,38,38,FF,69,39,39,FF,6
E,2E,2E,FF,BB,B4,3F,FF,BC,2C,3B,FF,BD,2D
,5F,9F,BE,2E,3A,FF,BA,7E,5E,FF,BF,3C,3E,
1C,C0,AB,BB,FF,DB,BA,AA,FF,DC,E7,1B9C
550 DATA C7,FF,DD,2B,2A,1B,DE,B4,60,1D,6
A,2A,2A,FF,6D,2D,2D,FF,6B,2B,2B,FF,FF,FF
,FF,FF,01,FF,1B,31,32,33,34,35,36,37,38,
39,30,BB,C0,08,09,51,57,45,52,54,14A8
560 DATA 59,55,49,4F,50,DD,DE,0D,11,41,5
3,44,46,47,48,4A,4B,4C,DC,DB,BA,10,BF,5A
,58,43,56,42,4E,4D,BC,BE,BD,10,6A,12,20,
14,70,71,72,73,74,75,76,77,78,79,12E9
570 DATA 90,91,24,26,21,6D,25,0C,27,6B,2
3,28,22,2D,2E,67,68,69,6D,64,65,66,6B,61
,62,63,60,6E,31,A6,A6,32,40,40,34,A3,A3,
51,A9,A9,57,AE,AE,38,7B,7B,39,7D,119A
580 DATA 7D,45,BC,BC,52,BD,BD,54,BE,BE,5
9,B0,B0,55,B9,B9,BF,5C,5C,49,B2,B2,4F,B3
,B3,50,B5,B5,DD,5B,5B,DE,5D,5D,FF,FF,FF,
61,E0,E1,E2,E3,65,E8,E9,EA,65,69,1DE6
590 DATA EC,ED,69,69,6F,F2,F3,F4,F5,75,F
9,FA,75,75,79,FF,FD,FF,FF,41,C0,C1,C2,C3
,45,C8,C9,CA,45,49,CC,CD,49,49,4F,D2,D3,
D4,D5,55,D9,DA,55,55,59,FF,DD,FF,20D0
600 DATA FF,20,60,B4,5E,7E,FF,FF,FF,FF,F
F,00,DC,00,08,00,08,00,FF,FF,07,08,09,FF
,04,05,06,FF,01,02,03,D1F,5555
```

MEMÓRIA

A memória é um dos componentes básicos de um computador. É utilizado pelo processador de duas maneiras: zona onde os dados de um programa estão guardados, podem ser lidos, alterados e escritos; zona onde o programa está e onde, sequenciamente é lido pelo processador, instrução após instrução seguindo uma sequência definida pelo próprio programa.

ESTANDO pois o programa em memória e a maioria dos dados também, (há dados que o processador pode receber/fornecer directamente de uma entrada/saída) a grande maioria dos acessos do processador estão ligados à memória. Daí que a velocidade de acesso a esta determine normalmente a velocidade de processamento do computador pela negativa (ou seja, o aumento da frequência de trabalho de um processador só tem efectivamente resultados equivalentes se este for acompanhado por circuitos de memória igualmente mais rápidos ou que, pelo menos, a sua lentidão relativa não elimine o ganho em velocidade de processamento). A tecnologia utilizada actualmente implica que os circuitos de memória sejam tanto mais caros quanto mais rápidos. Com as elevadas capacidades de memória que equipam os computadores actuais a diferença de preço torna-se significativa daí que se recorra frequentemente a uma hierarquização de memórias diferentes.

A memória de um computador é um conjunto de células em que cada uma pode ser endereçada inequivocamente (a cada endereço gerado corresponde uma e apenas uma célula) cujo conteúdo pode ser lido ou escrito pelo processador e que se mantém inalterado entre duas operações de escrita.

REGISTOS

Nem toda a memória se encontra no exterior do processador. Existem algumas células no seu interior que servem para guardar dados, normalmente de forma temporária, operandos e resultados de operações: são os registos internos do processador, não têm qualquer problema de velocidade de transferência mas são sempre em número muito limitado.

O preço de um processador é inflaccionado pelo número de registos internos disponíveis.

1 — MEMÓRIA CLÁSSICA

Até aos meados dos anos 70 os computadores obedeciam a uma arquitectura definida por John von Neuman. A memória era um "array" contínuo de células que podiam ser usadas indistintamente para guardar o programa e os dados. O desenvolvimento dos semicondutores permitiu a implementação de novas arquitecturas.

2 — MEMÓRIA SEGMENTADA

Este "ARRAY" é um suporte único, contínuo e indiferenciado para as informações acessíveis pelo processador quer elas sejam dados (individualmente ou em grupos) ou constituam uma sequência de instruções (programa). Tal permite uma grande economia de memória mas é-lhe inerente um grande risco: a não separação de programas e dados permite que, ao mais leve descuido qualquer uma das áreas seja invadida pela outra com os resultados desastrosos daí decorrentes.

Uma forma de ultrapassar este risco foi dividir a memória total em blocos específicos e com uma função determinada com um comprimento máximo determinado que, obviamente utilizando mais memória (há normalmente um desperdício) impede a interpenetração de zonas de dados e de programas (e mesmo de zonas diferenciadas de dados — por exemplo dados genéricos e pilhas). Tais blocos que, normalmente são recolocáveis em memória (podem situar-se em qualquer zona de memória total disponível) denominam-se segmentos.

Este arranjo lógico de memória é sobretudo útil em computadores multiutilizador pois que os endereços dentro de cada segmento são inalteráveis qualquer que seja a posição que este segmento ocupe na memória total do computador. Assim o mesmo programa pode estar a ser utilizado independentemente por vários utilizadores sem que tenha havido qualquer ajuste do proces-

sador bastando que cada "cópia" esteja localizada num segmento diferente dos outros.

Os computadores AMSTRAD PC utilizam este tipo de memória pois o seu processador é o 8086 da Intel. A óptima utilização destes recursos existe na utilização do AMSTRAD na configuração multiposto possível com o sistema operativo PROLOGUE.

3 — MEMÓRIA PAGINADA

Nos últimos anos o preço dos circuitos de memória tem visto o seu preço baixar significativamente (a recente subida de preços cremos ser um fenómeno transitório) contudo, conforme se disse no princípio existe uma diferença substancial entre memórias lentas e memórias rápidas (Nota: O conceito lento e rápido é muito subjectivo pois as memórias lentas de hoje são muito mais rápidas que as mais rápidas memórias de há uns anos atrás). Por isso é muito atractivo utilizar memórias hierarquizadas em que os dados e programas em utilização pelo processador estejam guardados em memórias rápidas e a restante informação em memórias lentas. Isto permitiria um processamento muito mais rápido sem ue tal implicasse um aumento grande de preço.

Para que o processador, contudo tenha acesso a toda a informação tem de se permitir que seja possível carregar na memória rápida qualquer bloco existente na memória lenta. Estes blocos chamam-se páginas e toda a informação é dividida em páginas de tamanho fixo.

Normalmente a memória lenta é constituída por discos magnéticos de alta capacidade enquanto a memória rápida é a RAM do sistema.

Inicialmente a primeira página é carregada na memória rápida e inicia-se o processamento.

Inevitavelmente (a menos que se trate de um programa muito simples e muito curto) nesta página existe uma referência de dados ou de programa (salto ou chamamento de sub-rotina) a uma outra página ainda não colocada na memória rápida e, como tal deve ser removida para esta.

Inevitavelmente, em determinada altura esta memória ficará cheia e uma página terá de ser removida desta para a memória mais lenta, a fim de ser substituída por outra requerida pelo processador. Para que esta substituição seja fácil e eficiente é necessário que este tipo de memória não exista, que as páginas sejam consecutivas. Para garantir isto, se estas páginas contiverem códigos de programas (que são por definição contínuos e contíguos) o endereçamento absoluto é transformado em dois endereços: endereço de página (guardado na tabela de páginas — é este endereço que determina a necessidade de substituição de página) e o deslocamento dentro desta ou seja o endereço relativo.

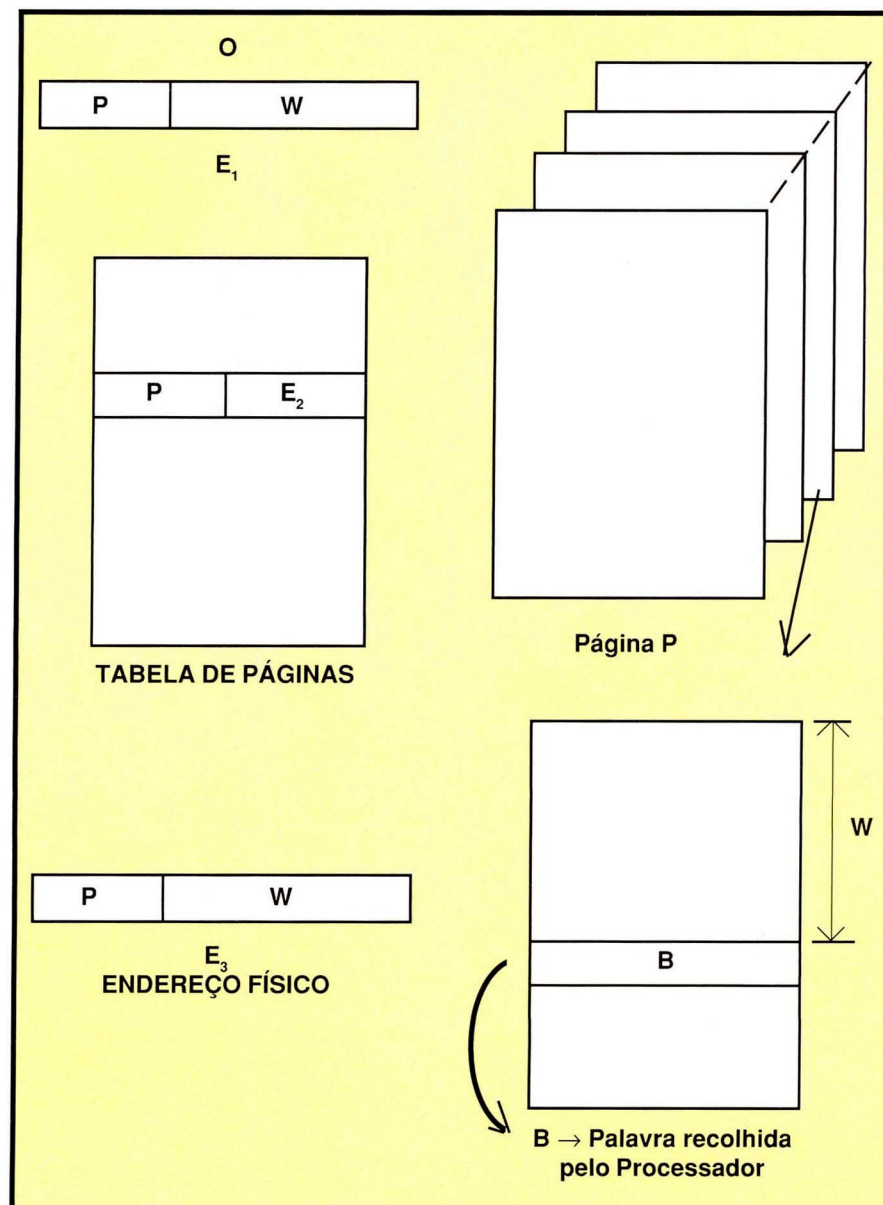



Figura M1

Para que o processador não gaste muito tempo a guardar na memória lenta páginas (que eventualmente pode vir a necessitar passado algum tempo) não deve haver nunca necessidade de libertar todas as páginas na memória rápida. Para isso é necessário que a gestão correcta das páginas de memória não implique que estas (sejam dados ou códigos) pertençam todas ao mesmo programa, para que o terminar de um programa não implique que todas as suas páginas sejam repostas na memória lenta. Deve ainda ser garantido, para optimização do tempo de processamento que as páginas mais utilizadas devam ser mantidas na memória rápida. Esta optimização implica ainda que só seja escrita de novo na memória lenta uma página que tenha sido alterada. Normalmente existe um indicador no descritor da página que indica ao processador — quando este pretende substitui-la — se esta foi modificada ou não, contudo um mecanismo muito simples pode aproximar-se muito deste, baseado no princípio de que só são possíveis de alteração as páginas de dados, nunca as de programa pelo que estas últimas nunca são re-escritas. Evidentemente que todo este processo tem de garantir a correcta re-escrita das páginas pelo que deve possuir uma tabela dos endereços em memória lenta das páginas em utilização na memória rápida.

4 — MEMÓRIA VIRTUAL

Utilizando a técnica hierárquica de memória paginada o processador pode endereçar um espaço de memória muito superior àquele a que está directamente conectado bastando que esse espaço exista algures, em forma de páginas e que, um mecanismo de paginação garanta que é transferido para memória directamente conectada ao processador, sempre que este lhe fizer referência, transformando endereços globais (virtuais) em endereços de página + deslocamento. Tal mecanismo denomina-se memória virtual pois o espaço total de memória tal qual é endereçado pelo processador não existe na realidade.

O processador do AMSTRAD PC 2286 (o Intel 80286) assegura já este mecanismo desde que trabalhe no modo protegido (a mémória endereçada directamente pode estender-se até 16 megabytes enquanto que a memória virtual pode atingir 1 gigabytes.

Por sua vez o processador do AMSTRAD 2386 (o 80386) faz também gestão de memória virtual mas em dois níveis: 1 — A memória real (que se pode, teoricamente estender até 4 gigabytes) pode ser manipulada pelo processador através de uma memória cache (ultra rápida — 35 ns de tempo de acesso) de 64 Kb podendo implementar-se um espaço de memória virtual até 64 terabytes (2^{64} bytes).

Esta mobilidade de páginas que, tal como na memória segmentada podem ser protegidas (páginas de dados diferentes de páginas de programa), permite a este método tratar com módulos (programa ou dados), de qualquer tamanho sem procedimentos especiais, contrariamente ao que se passa na memória segmentada cujo procedimento normal se resume a módulos de tamanho inferior ao do segmento. Módulos de tamanho superior ao do segmento implicam mecanismos complexos de endeçamento principalmente se estes módulos forem módulos de programa.

O mecanismo da paginação supõe sempre dois tipos de memória (a memória primária e a memória secundária) com a primeira muito mais rápida e inferior em tamanho que a segunda.

O processador trabalha sempre directamente sobre a primeira mas refere-se-lhe como se esta fosse a segunda. Como tal, possui uma capacidade de endereço muito superior à memória sobre a qual directamente trabalha. A totalidade da memória "endereçável" pelo processador chama-se pois memória virtual.

Dado ser a memória primária de tamanho muito inferior ao da memória secundária, o seu endereço é radicalmente diferente do endereço gerado pelo processador — endereço virtual.

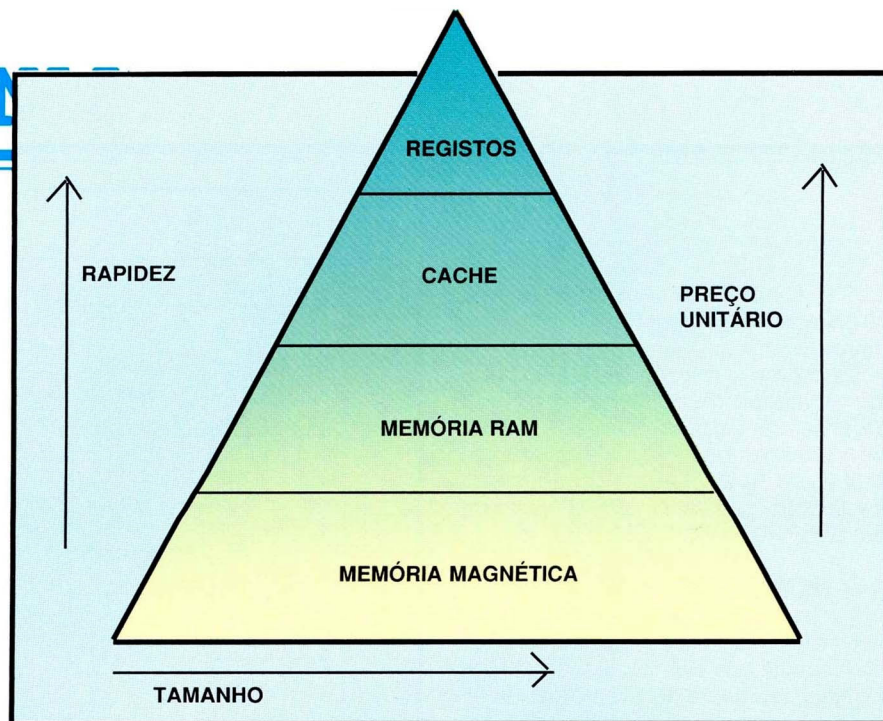



Figura M2

Por isso, o mecanismo de gestão das páginas tem de executar uma translação dinâmica de endereço (também chamada mapeamento de memória ou relocação de endereço) para que a determinado endereço virtual corresponda o correcto endereço físico. Esta translação é chamada dinâmica pois não há nenhuma relação constante no tempo entre estes dois endereços. O endereço físico associado a um endereço virtual depende das páginas existentes em memória ao momento e da sua localização. Se, entre páginas existentes em memória ao momento e da sua localização. Se entre duas referências ao mesmo endereço virtual houve movimentação da página entre a memória primária e a memória secundária é mais que provável que o segundo endereço físico seja substancialmente diferente do primeiro.

Todos os endereços virtuais são convertidos em endereços reais por consulta de uma tabela (que é dinâmica). Se desta consulta resultar que a página pretendida não está na memória primária, o mecanismo de translação provoca uma interrupção (instrution abort) sinalizando o CPU (ou o gestor da memória).

Este, analizada a causa da interrupção, vai transferir a página da memória secundária para a memória primária (tendo libertado o espaço necessário nesta, caso estivesse cheia). Este mecanismo é chamado "SWAPPING".

Todo este mecanismo é, contudo transparente para os programas quaisquer que eles sejam que podem dispôr de toda a memória virtual endereçável pelo gestor de memória independentemente da memória física instalada no board do processador, e directamente conectada a este. Realce-se ainda que, sendo este processo dinâmico, esta memória pode ser muito inferior à memória directamente conectável ao processador.

De novo, como exemplo, o AMSTRAD PC 2386 é fornecido em standard com 4 Mbytes e, o seu processador (o 80386) pode endereçar nestes os 64 terabytes, embora a memória que se pode ligar directamente e fisicamente endereçável vá até aos 4 gigabytes. Daí que as necessidades de memória física num sistema tenham a ver só com a rapidez pois que, quanto menos esta for maior número de SWAPPINGS são necessários.

A figura M1 mostra de uma forma simples o mecanismo de endereçamento físico de uma memória virtual paginada.

O endereço E1 é o endereço virtual, composto por P e W sendo P o número da página e W o deslocamento dentro desta. O mecanismo de gestão de memória virtual começa por separar neste o número da página. Com este valor "vai percorrer" a tabela de páginas. Se esta existir na memória primária terá associada a si, nesta tabela o endereço E2, com o deslocamento W (segundo elemento do endereço virtual) encontra-se B que é a palavra virtualmente endereçada sendo pois o elemento de memória que o processador recolherá.

O endereço físico é E3 constituído por E2 (endereço de início de página) e W (deslocamento na página).

Quando é necessário substituir páginas na memória primária há duas questões que se colocam:

1 — Qual a página a retirar de memória primária?

2 — Qual (ou quais) a(s) página(s) a ler na memória secundária?

Para resolver estas questões há vários métodos mas, em qualquer dos

casos, há três fundamentais para o primeiro e dois para o segundo

1 — Primeira página a retirar

A — LRU (Least Recently Used — A menos usada recentemente)

B — LFU (Least Frequently Used — A menos frequentemente usada)

C — FIFO (First In First Out — A primeira a entrar é a primeira a sair).

2 — Primeira página a ler

A — Por pedido (é lida apenas a página referida e não existente na memória primária)

B — Por anticipação (utiliza-se o princípio da localidade. Este princípio diz que, em informática há um "Working Set" localizado ou seja, a maioria das referências quer a dados quer a programas (saltos e chamamentos de subrotinas) se situa nas imediações dos dados actualmente em serviço e do programa actualmente a ser executado. como tal se há uma falta actual de uma página as faltas imediatas previsíveis serão de páginas adjacentes a esta. Assim em vez de se transferir apenas uma página transferem-se várias páginas anteriores e posteriores (se não estiverem já na memória primária, evidentemente) a esta. Assim, por antecipação está a fazer-se uma transferência que embora mais demorada em si, vai evitar transferências futuras diminuindo genericamente o tempo de acesso global à memória.

5 — MEMÓRIA CACHE

O aumento da frequência de trabalho dos processadores veio trazer a necessidade da utilização de memória também muito mais rápidas para que aquele não tivesse de esperar (utilizando ciclos "mortos — Wait States) cada vez que fizesse um acesso à memória conectada ao seu barramento. Contudo, como atrás já foi referido, estas memórias são muito mais caras que as suas congéneres mais lentas. Analogicamente com o sucedido entre a memória magnética e a memória RAM, a solução é a hierarquização desta última ficando a memória mais rápida conhecida como memória CACHE.

Os sistemas com memória CACHE (como o AMSTRAD 2386) possuem pois dois níveis de hierarquização em três tipos de memória (a CACHE, a RAM (genérica) e o Disco), absolutamente transparentes para o utilizador que pode aceder aos 4Mb de Ram do sistema como se todos eles fossem os 64 Kbytes de CACHE de 35 nanosegundos e aos 65 Mbytes de disco como se fossem os 4 Mb de Ram resultando um largo espaço de endereço (limitado apenas pelo tamanho máximo possível de um ficheiro em disco) com apenas 0,5 estados de "Wait" (em média, evidentemente).

Tal como foi visto anteriormente, também a gestão da memória virtual com CACHE implica a divisão da memória RAM em páginas de valor fixo que são transferidas para CACHE sempre que haja do CPU a necessidade de uma palavra (instrução ou dado) que não esteja ainda nesta última, obedecendo a um algorítmo de substituição. O endereçamento para memória CACHE também, identicamente, sofre uma translação dinâmica, em endereço de página + deslocamento. É pois idêntico ao que atrás foi dito sobre memória virtual sendo a única diferença o tempo de acesso (a memória primária é aqui mais rápida) e o tamanho da página (o tamanho da página transferida para a memória CACHE é menor — normalmente uma fracção da página transferida do disco para a memória RAM genérica).

Para terminar a figura M2 mostra duma forma esquemática a relação entre tamanho e rapidez das diferentes memórias.

Mário Leite

DIGITALIZAÇÃO DE SONS

Como é do conhecimento geral, os computadores são máquinas preparadas para o tratamento numérico da informação, isto é máquinas que apenas são capazes de tratar números. A esta informação numérica, dá-se o nome de "informação digital".

No entanto, no mundo real, a maior parte das informações e dados com que trabalhamos não são números nem se podem quantificar com facilidade. É o que chamamos informação analógica. Um bom exemplo, são as imagens, constituídas por linhas rectas e curvas, que dificilmente podem ser tratadas de forma numérica como é exigido pelos sistemas informáticos.

Ao processo de transformação de uma informação analógica no seu equivalente digital, chama-se digitalização e este é um dos campos mais apaixonantes da informática.

Neste artigo, vamos concentrar-nos na digitalização de sons, deixando para outra ocasião a digitalização de imagens. Começamos por dar uma pequena definição de som e, a partir daí veremos como é possível transformá-lo em números.

Um som não é senão uma onda, mais ou menos complicada, de amplitude e frequência variáveis. Pois bem, o que se faz para digitalizar a dita onda é dividi-la em pequenos troços iguais, considerando o valor médio da amplitude da onda em cada troço. Estes valores são tudo o que necessitamos para conter toda a informação sobre a onda e, assim, poder reproduzi-la posteriormente. O valor de amplitude considerado é escalonado em relação a um número que tomamos como valor máximo de amplitude. Esse número será dado em função da resolução que quisermos; por exemplo, se dispusermos de oito bits para armazenar os valores de amplitude, temos então, 256 valores possíveis.

Num computador de oito bits, em cada byte, pode guardar-se um valor de amplitude dos 256 possíveis; com estes valores é possível obter uma qualidade de reprodução mais do que aceitável.

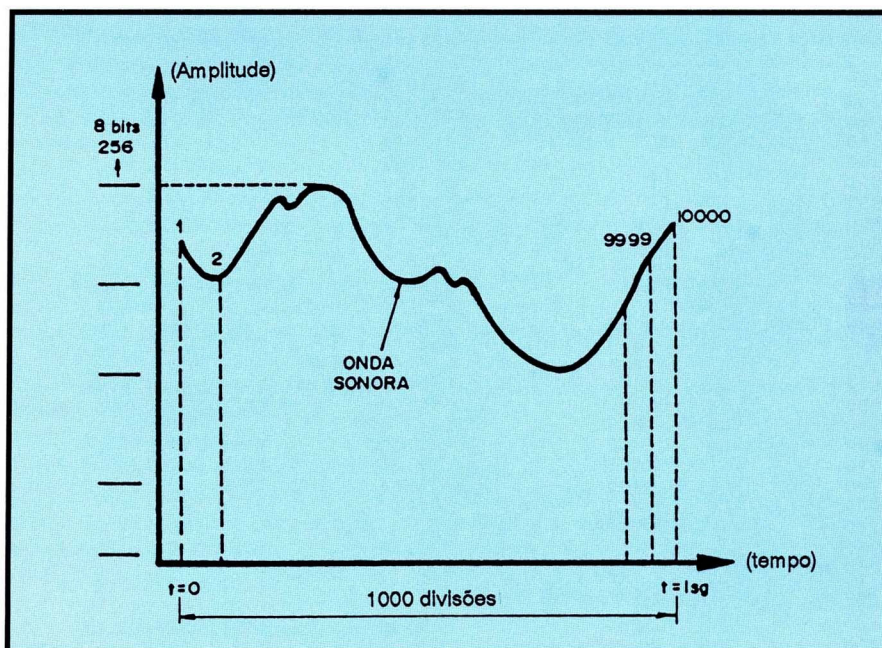
A qualidade do som depende pois, fundamentalmente, de dois factores. Por um lado, do número de valores em que podemos dividir a amplitude e, por outro, da velocidade de, ou, dizendo de outra forma, a velocidade a que dividimos a onda para medir o seu valor de amplitude. Este valor da velocidade de amostragem tem de ser estabelecido, de forma a que exista equilíbrio entre a qualidade do som e a duração do mesmo que conseguimos reter em memória. É evidente que, se tomarmos 10000 pontos de amostras num segundo de onda, conseguiremos obter fidelidade capaz de ser reproduzida. Contudo, utilizariamos 30 K bytes de memória para armazenar apenas três segundos de sons. Assim, fica clara a necessidade de um compromisso entre a duração do registo sonoro e a sua qualidade.

Os factores que vimos até agora permitem-nos explicar de que forma podemos digitalizar um determinado som no nosso CPC. Em primeiro lugar, necessitamos de conhecer a forma de introduzir um sinal sonoro no computador e que este o vá transformando em números. Para isso vamos utilizar o leitor de cassettes, porque permite a introdução fácil da música obtida a partir de qualquer fita gravada.

Quando o computador lê, através da entrada do leitor de cassettes um programa qualquer, o que faz é transformar uma série de sons de determinada frequência e duração em 0 e 1 ; pois bem, vamos empregar esta técnica para desenhar gráficamente a música. Empregamos apenas um bit para guardar o valor da amplitude da onda. Se a amplitude for superior a um certo valor, corresponderá a 1; se for inferior corresponderá a 0. Como se pode observar, a resolução é bastante pobre e o ruído que ouviremos ao reproduzir a música será bastante elevado. Contudo, este procedimento facilitar-nos-á bastante as coisas e os resultados obtidos serão

mais aceitáveis. Podendo guardar o valor da amplitude do sinal num só bit, poderemos realizar um ponto de gráfico a uma velocidade muito maior e, para além disso, gravar uma maior quantidade de som na memória. Num byte, poderemos guardar até oito pontos de gráfico e a reprodução da frequência da onda será muito boa.

Como vemos, a técnica é muito fácil: guardamos o som reproduzido pelo leitor de cassetes, durante um certo tempo, em 1 ou 0 , conforme a amplitude da onda seja grande ou pequena. Para reproduzir posteriormente o som ainda é mais fácil, pois ao ler os 0´s (zeros) e 1´s (uns) armazenados e, ao ir activando o altifalante está tudo feito. Pelo altifalante interno ouviremos uma reprodução digital do som gravado, ainda que, com algum ruído de fundo.

O programa que acompanha este artigo emprega a técnica, anteriormente, explicada para digitalizar o som. O que faz é criar alguns comandos **RSX**, explicados mais adiante que nos permitem realizar, facilmente, o trabalho.

Representação de um sinal sonoro com 8 bits de amplitude e 10 000 impulsos por segundo

```
 10 ;----------------------------
 20 ; AMSTRAD MAGAZINE - 1989
 30 ; COMANDOS RSX PARA DIGITALI-
 40 ; ZACAO E REPRODUCAO DE
 50 ; SONS
 60 ; ............
 70 ;----------------------------
 80 ;
 90 LOGRSX: EQU  #BCD1
100 TAPEON: EQU  #BC6E
110 TESTK:  EQU  #BB1E
120 RESET:  EQU  #BCA7
130 OUTCHR: EQU  #BB5A
140 TAPEOF: EQU  #BC71
150 ;
160 BEEP:   EQU  #07
170 CR:     EQU  13
180 LF:     EQU  10
190 VIDINV: EQU  #18
200 FIM:    EQU  #00
210 ;
220         ORG  #9000
230         LD BC,QUADRO
240         LD   HL,WRKSPC
250         CALL LOGRSX
260         RET
270 QUADRO: DEFW NOME
280         JP   RECORD
290         JP   RECORD
300         JP   PLAY
310         JP   PLAY
320         JP   COPYR
330 WRKSPC: DEFS 4
340 NOME:   DEFB "G"+#80
350         DEFM "GRAV"
360         DEFB "A"+#80
370         DEFM "REPRODU"
380         DEFB "Z"+#80
390         DEFB "R"+#80
400         DEFM "C"
410         DEFB "R"+#80
420         DEFB FIM
430 RECORD: LD   (STACK),SP
440         CP   #03
450         JP   NZ,SEGUE1
460         LD   A,(IX+0)
470         CP   #00
480         JP   Z,ERRO
490         LD   (RATE),A
500         INC  IX
```

```
 510         INC  IX
 520         JP   SEGUE2
 530 SEGUE1: CP   #02
 540         JP   NZ,ERRO
 550         LD   A,CR
 560         LD   (RATE),A
 570 SEGUE2: LD   E,(IX+0)
 580         LD   D,(IX+1)
 590         LD   L,(IX+2)
 600         LD   H,(IX+3)
 610         PUSH HL
 620         CALL TAPEON
 630         LD   HL,TEXTO3
 640         CALL PRINT
 650 CICLO0: LD   A,#42
 660         CALL TESTK
 670         JP   NZ,SAIDA
 680         LD   A,#2F
 690         CALL TESTK
 700         JP   NZ,SEGUE3
 710         JP   CICLO0
 720 SEGUE3: LD   HL,TEXTO2
 730         CALL PRINT
 740         PUSH DE
 750         CALL RESET
 760         POP  DE
 770         POP  HL
 780         CALL REC2
 790         LD   A,BEEP
 800         CALL OUTCHR
 810         CALL TAPEOF
 820         RET
 830 PLAY:   LD   (STACK),SP
 840         CP   #03
 850         JR   NZ,SEGUE4
 860         LD   A,(IX+0)
 870         AND  #0F
 880         LD   (VOLUME),A
 890         INC  IX
 900         INC  IX
 910         JR   SEGUE5
 920 SEGUE4: CP   #02
 930         JP   NZ,ERRO
 940         LD   A,#0F
 950         LD   (VOLUME),A
 960 SEGUE5: LD   E,(IX+0)
 970         LD   D,(IX+1)
 980         LD   L,(IX+2)
 990         LD   H,(IX+3)
1000         CALL PLAY2
```

```
1010         RET
1020 SAIDA:  LD   A,LF
1030         CALL OUTCHR
1040         CALL OUTCHR
1050         CALL TAPEOF
1060         LD   SP,(STACK)
1070         RET
1080 ERRO:   LD   HL,TEXTO1
1090         CALL PRINT
1100         RET
1110 PRINT:  LD   A,(HL)
1120         CP   #00
1130         RET  Z
1140         CALL OUTCHR
1150         INC  HL
1160         JP   PRINT
1170 REC2:   DI
1180         PUSH HL
1190         POP  IX
1200 CICLO1: LD   B,#08
1210         XOR  A
1220 CICLO2: CALL SCANIN
1230         CALL DORATE
1240         RLA
1250         DJNZ CICLO2
1260         LD   (IX+0),A
1270         INC  IX
1280         DEC  DE
1290         LD   A,D
1300         OR   E
1310         JR   NZ,CICLO1
1320         EI
1330         RET
1340 SCANIN: PUSH BC
1350         LD   B,#F5
1360         IN   H,(C)
1370         RL   H
1380         POP  BC
1390         RET
1400 DORATE: DEFB #26
1410 RATE:   DEFB #0D
1420 CICLO3: DEC  H
1430         JR   NZ,CICLO3
1440         RET
1450 PLAY2:  PUSH HL
1460         PUSH DE
1470         CALL PLAY3
1480         POP  DE
1490         POP  HL
```

CPC

```
1500 CICLO4: LD   B,#08
1510 CICLO5: RLC  (HL)
1520         PUSH BC
1530         CALL C,ACTIVA
1540         CALL NC,DESACT
1550         POP  BC
1560         DJNZ CICLO5
1570         INC  HL
1580         DEC  DE
1590         LD   A,D
1600         OR   E
1610         JR   NZ,CICLO4
1620         CALL #BCA7
1630         RET
1640 PLAY3:  CALL #BCA7
1650         DI
1660         LD   A,#03
1670         LD   C,#00
1680         CALL DOCHIP
1690         LD   A,BEEP
1700         LD   C,#3D
1710         CALL DOCHIP
1720         LD   A,#02
1730         LD   C,#00
1740         CALL DOCHIP
1750         RET
1760 ACTIVA: PUSH AF
1770         LD   A,#09
1780         LD   C,#0F
1790 VOLUME: EQU  $-1
1800         CALL DOCHIP
1810         POP  AF
1820         RET
1830 DESACT: PUSH AF
1840         LD   A,#09
1850         LD   C,#00
1860         CALL DOCHIP
1870         POP  AF
1880         RET
1890 DOCHIP: PUSH AF
1900         LD   A,#C0
1910         LD   B,#F6
1920         OUT  (C),A
1930         LD   B,#F4
1940         POP  AF
1950         OUT  (C),A
1960         LD   B,#F6
1970         LD   A,#80
1980         OUT  (C),A
1990         LD   B,#F4
2000         OUT  (C),C
2010         LD   B,#F6
2020         XOR  A
2030         OUT  (C),A
2040         RET
2050 COPYR:  LD   HL,TEXTO4
2060         CALL PRINT
2070         RET
2080 STACK:  DEFS 2
2090 TEXTO1: DEFB BEEP,LF,CR
2100         DEFM "ERRO NOS PARAME
             TROS"
2110         DEFB BEEP,LF,LF
2120         DEFB CR,FIM
2130 TEXTO2: DEFB BEEP,VIDINV
2140         DEFM "******** ESPERE
             POR FAVOR"
2150         DEFM " GRAVANDO *****
             ***"
2160         DEFB VIDINV,CR,LF
2170         DEFB LF,FIM
2180 TEXTO3: DEFB BEEP,CR,LF
2190         DEFM "PRIMA <ESPACO>
             PARA"
2200         DEFM " COMECAR A GRAV
             AR"
2210         DEFB CR,FIM
2220 TEXTO4: DEFB BEEP,CR,LF
2230         DEFB LF,#A4
2240         DEFM " A.M. 1989"
2250         DEFB CR,LF,LF
2260         DEFB BEEP,FIM
```

O primeiro dos comandos é |**GRAVA**, que se pode abreviar para |**G** e o que faz é armazenar a música introduzida pelo Leitor de Cassettes. Este comando dispõe de três parâmetros que são a localização de memória, a partir da qual podemos começar a armazenar os dados, o comprimento, e a velocidade.

```
10 REM *************************
20 REM * DIGITALIZADOR DE SONS *
30 REM * AMSTRAD MAGAZINE 1989 *
40 REM *************************
50 REM
60 REM
70 REM PARA GRAVAR A ROTINA FACA'
80 REM SAVE"DIGITA.BIN",B,&9000,&9300
90 REM
100 REM EXECUTAR ATRAVES DE: CALL &9000
110 REM
120 REM
130 REM DATA DO CODIGO MAQUINA
140 DATA 010A90211B90CDD1BCC91F90C33290C
33290C39890C39890C37E9100000000C7
150 DATA 47524156C1D2524550524F4455DAD24
3D200ED738591FE03C24D90DD7E00FE00
160 DATA CAD890321091DD23DD23C35790FE02C
2D8903E0D321091DD5E00DD5601DD6E02
170 DATA DD6603E5CD6EBC21D491CDDF903E42C
D1EBBC2C8903E2FCD1EBBC28090C36D90
180 DATA 21A291CDDF90D5CDA7BCD1E1CDEA903
E07CD5ABBCD71BCC9ED738591FE03200E
190 DATA DD7E00E60F325291DD23DD23180AFE0
2C2D8903E0F325291DD5E00DD5601DD6E
200 DATA 02DD6603CD1591C93E0ACD5ABBCD5AB
BCD71BCED7B8591C9218791CDDF90C97E
210 DATA FE00C8CD5ABB23C3DF90F3E5DDE1060
8AFCD0691CD0F911710F7DD7700DD231B
220 DATA 7AB320EAFBC9C506F5ED60CB14C1C92
60D2520FDC9E5D5CD3491D1E10608CB06
230 DATA C5DC4E91D45891C110F4231B7AB320E
CCDA7BCC9CDA7BCF33E030E00CD62913E
240 DATA 070E3DCD62913E020E00CD6291C9F53
E090E0FCD6291F1C9F53E090E00CD6291
250 DATA F1C9F53EC006F6ED7906F4F1ED7906F
63E80ED7906F4ED4906F6AFED79C921FE
260 DATA 91CDDF90C90000070A0D4552524F204
E4F53205041524D4554524F53070A0A
270 DATA 0D0007182A2A2A2A2A2A2A20504F522
04641564F52204553504552452054D20
280 DATA 4D4F4D454E544F202A2A2A2A2A2A2A1
80D0A0A00070D0A205052494D41203C45
290 DATA 535041434F3E205041524120434F4D4
5434152204120475241564152D00070D
300 DATA 0A0A2843292041 2E4D2E2D203139383
90D0A0A0700000000000000000000000
310 MODE 2:PRINT"CARREGAMENTO DA DATA."
320 MEMORY 36863:CHECK=0:FOR N=36864 TO
37404 STEP 32
330 READ A$:FOR I=0 TO 31
340 A=VAL("&"+MID$(A$,(I*2)+1,2)):POKE N
+I,A:CHECK=CHECK+A
350 NEXT:NEXT:IF CHECK<> 58543 THEN PRIN
T "ERRO NAS LINHAS DE DATA":END
360 CALL &9000
370 MODE 2:PRINT"COMANDOS PRONTOS A EXEC
UTAR."
```

Podemos utilizar as posições compreendidas entre &500 e &8FFF para armazenar o som. Se começarmos a armazenar o som a partir do endereço &500, teremos uns &8000 bytes livres de capacidade de memória ; isto, juntamente com a velocidade de armazenamento, inicialmente 13, permitir-nos-á obter uns 25 segundos de gravação. A velocidade de armazenamento, é um valor compreendido entre 0 e 255. Quanto mais elevado for o valor, mais lenta será a recolha de valores, pelo que poderemos armazenar maior quantidade de som, ainda que de inferior qualidade. A velocidade de armazenamento é controlada por um simples ciclo de espera que retarda o acto da leitura do Leitor de cassettes.

Para reproduzir o som gravado dispomos do comando |**REPRODUZ** que se abrevia em |**R**.. Este comando possui, igualmente três parâmetros : o primeiro é a localização do ínicio dos sons a reproduzir, o segundo corresponde ao comprimento dos dados a reproduzir e o terceiro é o volume de reprodução. Se quisermos reproduzir um som gravado, por exemplo, com o comando |**GRAVA, &500, &7000**, na sua totalidade, temos que accionar o comando |**REPRODUZ, &500, &7000**, Se, apenas, nos interessar parte do som, podemos variar os parâmetros de ínicio e de comprimento, conseguindo, assim, efeitos muito interessantes.

Se desejarmos guardar em fita a informação gravada, teremos apenas que executar a seguinte ordem:

SAVE "*nome do ficheiro*", B, &500, &7000 guardando, assim, em fita ou em disco todo o som digitalizado.

Resumindo os comandos, vamos apresentá-los com todos os seus parâmetros

|GRAVA, localização de ínicio, comprimento, velocidade de armazenamento

|REPRODUZ, localização de ínicio, comprimento, volume

Segue-se a listagem do programa, em **BASIC**, que deve ser introduzida no computador com algum cuidado afim de evitar possiveis erros, especialmente nas linhas de **DATA** nela incluidas.

Esperamos que a digitalização de sons seja um tarefa agradável e, que o vosso interesse contribua para melhorar o programa e, assim, obter uma maior fidelidade na reprodução dos sons.

BYTE INFORMÁTICA

CONJUNTO MONUMENTAL INFANTE — SALA 204
9000 FUNCHAL

— Porque queremos que a informática chegue a todos

— Compre agora o seu **AMSTRAD** pagando-o em 18 prestações.

— Prefira o revendedor autorizado **AMSTRAD.**

A RAZÃO DA ESCOLHA CERTA!

— Aplicações por medida.

— Aplicações normalizadas

Contabilidade
Stocks
Facturação
Contas Correntes
Fornecedores/Clientes
Salários
Vídeos

S.O.S. LUNA 2

VOCÊ é o comandante de um serviço de Assistência Intergaláctica e foi informado que quatro cientistas ficaram presos na superfície lunar. Salvá-los, é o seu dever. No entanto, só ao parar a sua nave sobre a lua, você se apercebe que, por baixo de si, uma perigosa nuvem de asteróides o separa dos cientistas em perigo.

Quando achar que é a altura própria deverá lançar a pequena nave de salvamento e guiá-la através da nuvem de asteróides para recolher um dos cientistas. Depois de recolher o cientista tenha, no entanto, muita atenção porque quando regressar à nave espacial a nuvem de asteróides se transforma... num enorme exército de discos voadores.

Ganhará dez pontos de cada vez que conseguir salvar um cientista.

Se, no entanto, colidir com um asteróide, com um disco voador ou com a superfície lunar, ou não conseguir alunar a nave espacial, perde uma vida.

No início de cada jogo o jogador tem direito a três vidas e 25 unidades de combustível, e com elas deve salvar quatro cientistas. Para animar a espinhosa tarefa que o espera, cada jogo é inicializado com uma pequena melodia que, esperamos, nada tenha a ver com a "melodia do desespero".

As teclas utilizadas são: Z = esquerda; X = direita; \ = lançar nave de salvamento/propulsão.

Não são necessárias quaisquer instruções de carregamento ou execução; introduza simplesmente o programa apresentado, grave-o, e execute-o como costuma fazer com todos os outros programas de BASIC.

```
5 MODE 0:PRINT "S.O.S. LUNA 2"
10 ENV 1,7,2,1:ENV 2,15,-1,8:ENV 3,7,-2,
1,7,2,1,7,-2,1,7,2,1
20 READ NO,DU:IF NO=-1 THEN 50
30 SOUND 1,NO,-DU,15,3
40 GOTO 20
41 DATA 159,3,213,1,190,3,213,1,119,1,12
7,1,142,1,159,1,142,3,213,1,119,1,127,1,
142,1,159,1,142,2,213,2,106,8
42 DATA 95,2,106,1,159,1,106,2,119,2,127
,2,190,1,159,1,127,2,142,2
43 DATA 127,4,119,1,127,1,142,1,159,1,14
2,4,127,1,142,1,159,1,169,1,159,8
49 DATA -1,-1
50 SYMBOL AFTER 97
60 FOR I=20000 TO 20048:READ A:POKE I,A:
NEXT
70 FOR I=20500 TO 20526:READ A:POKE I,A:
NEXT
80 FOR I=20550 TO 20576:READ A:POKE I,A:
NEXT
90 DATA &dd,&21,&30,&75,&06,&05,&dd,&7e,
&00,&3d,&fe,&ff,&c2,&31,&4e,&3e,&14,&dd,
&77,&00,&dd,&23,&10,&ee
100 DATA &dd,&21,&35,&75,&06,&05,&dd,&7e
,&00,&3c,&fe,&15,&c2,&49,&4e,&3e,&00,&dd
,&77,&00,&dd,&23,&10,&ee,&c9
110 DATA &06,&05,&dd,&21,&30,&75,&0e,&05
,&dd,&66,&00,&69,&cd,&75,&bb,&3e,254,&cd
,&5a,&bb,&dd,&23,&0c,&0c,&10,&ee,&c9
120 DATA &dd,&21,&35,&75,&0e,&06,&06,&05
,&dd,&66,&00,&69,&cd,&75,&bb,&3e,&ff,&cd
,&5a,&bb,&dd,&23,&0c,&0c,&10,&ee,&c9
130 FUEL=25:SCORE=0:VI=3:HOM=4
140 MODE 0:FUEL=25:INK 0,0:INK 2,2:INK 3
,9:INK 4,15:BORDER 0:INK 1,26:INK 5,24:I
NK 6,8,26:FOR I=30000 TO 30009:POKE I,IN
T(RND*20):NE
XT:WINDOW #1,1,20,4,14
150 SPEED INK 5,5
160 SYMBOL 254,56,126,127,255,255,255,12
4,56:SYMBOL 255,120,252,252,254,254,63,6
3,31
170 SYMBOL 97,0,0,7,15,31,63,127,255:SYM
BOL 98,0,0,255,165,165,255,255,129:SYMBO
L 99,0,0,&X11100000,240,248,252,254,255:
SYMBOL 100,2
55,243,243,255,255,127,63,31:SYMBOL 101,
255,&X11001111,&X11001111,255,255,254,25
2,248
180 SYMBOL 102,192,240,240,248,248,248,2
54,255:SYMBOL 103,1,15,31,31,31,127,255,
255:SYMBOL 104,255,255,255,255,255,255,2
55,255
190 SYMBOL 240,24,36,66,66,36,60,90,195
200 PEN 3:LOCATE 1,15:PRINT"f
       ghf                   ghhhf
      ghhhhhf              ghhhhhhhf
    ghhhhhhhh
hf          ghhhhh";
210 PRINT "hhhhhhhhhhhhhhhhhhhhhhhhhhhhh
hhhhhhhhhhh"
220 PEN 4:LOCATE 7,20:PRINT CHR$(250);"
";:IF HOM>1 THEN PRINT CHR$(248);"  ";:I
F HOM>2 THEN PRINT CHR$(248);" ";:IF HOM
>3 THEN PRIN
T CHR$(251)
230 A=1:D=1
240 REM
250 IF D=1 AND A<16 THEN A=A+1 ELSE D=0
260 IF D=0 AND A>1 THEN A=A-1 ELSE D=1
270 PEN 2:LOCATE A,1:PRINT " abc ":LOCAT
E A,2:PRINT " d e "
280 PEN 1:CALL 20000:CLS #1:CALL 20500:C
ALL 20550
290 IF INKEY(22)=0 THEN SOUND 1,120,12,0
,1:GOTO 310
300 GOTO 250
310 X%=A*32+32:Y%=400-33
320 REM ******* CICLO PRINCIPAL *******
330 CALL 20000:CLS #1:PEN 1:CALL 20500:C
ALL 20550
340 IF INKEY(71)=0 AND FRENTE=0 AND X%>0
 THEN GOSUB 450:X%=X%-32
350 IF INKEY(63)=0 AND FRENTE=0 AND X%<6
07 THEN GOSUB 450:X%=X%+32
360 IF INKEY(22)=0 AND FUEL>0 THEN SOUND
 1,0,25,7,0,0,2:FUEL=FUEL-5:FRENTE=1 ELS
E FRENTE=0
370 IF FRENTE=0 THEN GOSUB 450:Y%=Y%-16
380 CH=TEST (X%+15,Y%-7)
390 IF CH=3 THEN 460
400 PLOT X%,Y%,5:TAG:PRINT CHR$(240);:TA
GOFF
410 IF CH=1 THEN 460
420 IF CH=4 THEN 540
430 LOCATE 1,23:PEN 7:PRINT "FUEL:";FUEL
440 GOTO 330
450 MOVE X%,Y%:TAG:PRINT " ";:TAGOFF:RET
URN
460 PLOT X%,Y%,6:TAG:PRINT CHR$(238);:TA
GOFF
470 SOUND 1,0,0,15,2,0,7
480 LOCATE 1,24:PEN 8:VI=VI-1:PRINT"VIDA
S:";VI:FOR I=1 TO 500:NEXT
490 IF VI=0 THEN 500:ELSE 140
500 LOCATE 6,10:PEN 1:PRINT "GAME OVER"
510 LOCATE 6,11:PRINT "SCORE:";SCORE
520 LOCATE 6,13:PRINT "SPACE ..."
530 IF INKEY(47)<>0 THEN 530 ELSE RUN
540 INK 1,6:SYMBOL 254,0,60,126,165,165,
126,60,0:SYMBOL 255,0,60,126,165,165,126
,60,0
550 REM ********** CIMA **********
560 PEN 1:CALL 20000:CLS #1:CALL 20500:C
ALL 20550
570 GOSUB 450
580 IF INKEY(71)=0 AND X%>0 THEN X%=X%-3
2
590 IF INKEY(63)=0 AND X%<607 THEN X%=X%
+32
600 Y%=Y%+16
610 CH=TEST (X%+15,Y%-2)
620 PLOT X%,Y%,5:TAG:PRINT CHR$(240);:TA
GOFF
630 IF CH=1 THEN 460
640 IF Y%>368 THEN IF X%<>A*32+32 THEN 4
60 ELSE 660
650 GOTO 560
660 SOUND 1,60,25,15:HOM=HOM-1:SCORE=SCO
RE+10:IF HOM=0 THEN 670 ELSE 140
670 HOM=4:GOTO 140
```

CRASH! é uma adaptação para CPC 6128 do conhecidissimo jogo ´The Wall´.

Escolhemos este jogo, porque, na nossa opinião, é dos jogos mais fáceis de jogar; e como tal, um dos mais solicitados não só por quem tem o seu primeiro contacto com os computadores, mas também por todos os outros que embora possam achá-lo um jogo simples, acabam por se prender a ele ´horas e horas a fio´.

Para quem não conhece ainda este maravilhoso jogo aí vai a explicação: o jogador controla um martelo que lhe permite partir uma série, variável, de tijolos; por cada vez que premir uma tecla. Ao fim de cinco jogadas, a parede aproxima-se, perigosamente, de sí com mais uma fileira de tijolos. Aqui, no entanto, tem de ter em consideração que se deixar o martelo percorrer todo o ecran sem lhe ´tocar´ o computador conta mais uma jogada. O objectivo do jogo é, portanto, como já puderam adivinhar, destruir a parede permanecendo o máximo de tempo em jogo. O jogo termina quando o martelo chocar com a parede lateralmente. Mas, para mais pormenores utilize a opção 2 do jogo.

ESTRUTURA DO PROGRAMA:

100 - 130	Ecrã do jogo
200 - 290	Passagem do martelo
300 - 350	Martelada
400 - 420	Pontuação
500 - 510	Verifica se bateu no tijolo, lateralmente
590	Acrescenta mais uma carreira de tijolos
600 - 732	Fim
800 - 840	Instruções
900 - 930	Hi-Score
950 - 999	Menu

VARIÁVEIS PRINCIPAIS:

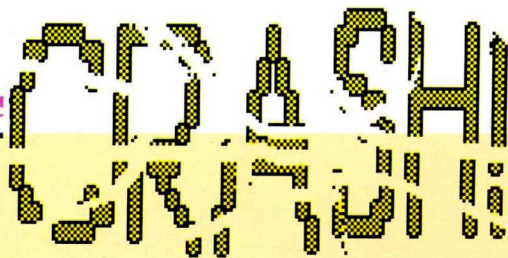
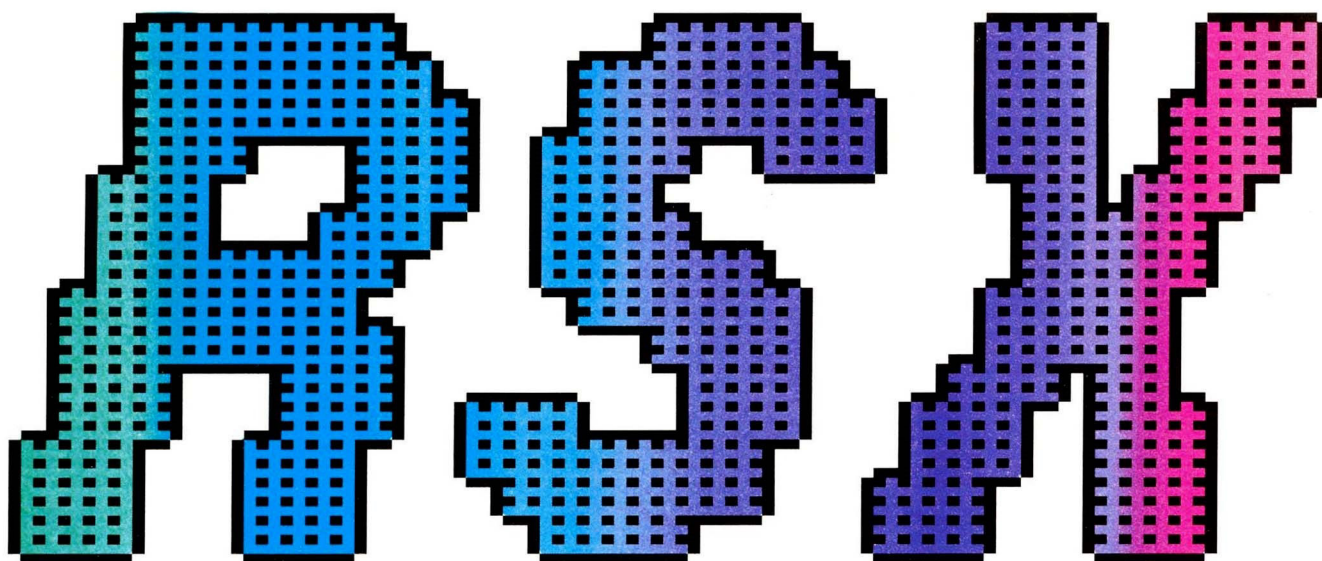
SC Score
HI Hi-Score
HI$ Nome do Jogador que obteve o Hi-score

ALTERAÇÕES POSSÍVEIS:

Martelo mais rápido - retirar linha 231
Nº de tijolos destruídos por jogada - alterar linha 310 para

FOR X=1 TO (Nº): ...

Resta-me portanto desejar-lhes bons momentos testando os reflexos com CRASH !

Jorge Prata

```
1 ' AM & J.PRATA (C) 1989
2 MODE 1:PEN 3:PAPER 0:BORDER 1
5 U$="000000":HI$="J.PRATA":HI=0:W$="000
000"
10 SYMBOL AFTER 190
20 SYMBOL 190,255,171,213,171,213,171,21
3,255
30 SYMBOL 191,8,28,14,15,18,32,64
40 SYMBOL 192,0,0,0,6,6,126,6,6
50 GOTO 950
100 ' ECRAN INICIAL
105 SC=0
110 A$="":FOR F=1 TO 39:A$=A$+CHR$(190):
NEXT
120 CLS:LOCATE 1,11:FOR F=0 TO 11:PRINT
A$:NEXT
130 PRINT:PRINT:M$="SCORE:000000 HI-SCOR
E:"+U$+" -> "+HI$:PEN 1:PRINT M$
200 ' JOGADA
210 PEN 2:FOR Z=1 TO 5
220 FOR N=1 TO 39
230 LOCATE N,1:GOSUB 500:PRINT CHR$(191)
:SOUND 1,N,0.5,8
231 FOR H=1 TO 10:NEXT
240 LOCATE N,1:PRINT " "
250 IF INKEY$<>"" THEN GOTO 300
260 NEXT N
265 NEXT Z
270 PEN 3:LOCATE 1,23:PRINT A$:PRINT:PRI
NT "
  ":PEN 1:PRINT M$:GOSUB 420
290 GOTO 210
300 ' MARTELADA
310 FOR X=1 TO ((RND*14)+10):U$=INKEY$:L
OCATE N,X:GOSUB 410:PRINT CHR$(191)
312 FOR H=1 TO 10:NEXT
320 LOCATE N,X:PRINT CHR$(192)
322 FOR H=1 TO 10:NEXT
330 LOCATE N,X:PRINT " "
340 NEXT X
350 GOTO 265
400 ' PONTOS
410 IF COPYCHR$(#0)=CHR$(190) THEN SOUND
 1,10,1,12: SC=SC+1
420 PEN 1:D$=STR$(SC):D$=RIGHT$(D$,(LEN(
D$)-1)):D=6-LEN(D$):LOCATE 7,25:PRINT LE
FT$(W$,D);D$:LOCATE N,X:PEN 2:RETURN
500 IF COPYCHR$(#0)=CHR$(190) THEN GOTO
600
510 LOCATE N,1:RETURN
590 FOR G=1 TO 39:PRINT CHR$(190);:NEXT
G:RETURN
600 ' FIM
602 PEN 3:LOCATE 1,23
610 FOR J=1 TO 5:GOSUB 590:PRINT:NEXT J
620 RESTORE:FOR I=1 TO 8:FOR T=1 TO 39:R
EAD K:IF K=0 THEN PRINT CHR$(190);
630 IF K=1 THEN PRINT " ";
640 NEXT T:PRINT:NEXT I
650 DATA 0,0,1,1,1,1,1,0,1,1,1,1,1,0,0,0
,1,1,1,1,0,0,0,1,1,1,1,1,0,1,0,0,0,0,1,0
,0,1,0
660 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0
,0,1,0
670 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0
,0,1,0
680 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0
,0,1,0
690 DATA 0,1,0,0,0,0,0,0,1,1,1,1,1,0,0,1
,1,1,1,1,1,0,0,1,1,1,1,0,0,1,1,1,1,1,1,0
,0,1,0
700 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0
,0,1,0
710 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0
,0,0,0
720 DATA 0,0,1,1,1,1,1,0,1,0,0,0,0,1,0,1
,0,0,0,0,1,0,1,1,1,1,1,0,0,1,0,0,0,0,1,0
,0,1,0
730 FOR F=1 TO 7:GOSUB 590:PRINT:NEXT F
732 PEN 1:GOTO 900
800 ' INSTRUCOES
802 PEN 2
805 MN$="   Aqui o objectivo e manter-se
 o mais  tempo possivel em jogo. Assim,
tem de   destruir os tijolos, antes que
estes    che
guem ao nivel do martelo."
810 MM$="   Ao gravar atraves da opcao 2
, pode   manter o jogo tal como o deixou
, apos a ultima utilizacao. Para tal dev
e carre- gar
 o jogo fazendo: LOAD 'CRASH!' e, em seg
uida GOTO 920."
812 TT$="   CONTROLOS=?(a tecla que bem
entender)"
815 NN$="AM & J.PRATA DESEJAM-LHE BONS R
EFLEXOS.."
820 CLS:PEN 3:LOCATE 16,4:I$="InStRuCoEs
":PRINT I$:PEN 2:PRINT:PRINT:PRINT MN$:P
RINT:PRINT MM$:PRINT:PRINT:PEN 1:PRINT T
T$:PRINT:PRI
NT:PEN 3:PRINT NN$
830 PEN 3:LOCATE 16,4:PRINT I$
832 FOR F=1 TO 40:NEXT
834 SOUND 1,(RND*100),1,6
840 LOCATE 16,4:PRINT "iNsTrUcOeS":IF IN
KEY$="" THEN GOTO 830
900 ' MENSAGENS
910 IF SC>HI THEN HI=SC:PRINT:PRINT:LOCA
TE 1,25:PRINT "PARABENS CONSEGUIU BATER
O HI-SCORE...":PRINT:PRINT:GOTO 920
912 GOTO 950
920 PEN 2:INPUT "NOME (Max.7 Ltrs)";N$:H
I$=LEFT$(N$,7)
930 D$=STR$(HI):D$=RIGHT$(D$,(LEN(D$)-1)
):D=6-LEN(D$):U$=LEFT$(W$,D)+D$
950 CLS:PEN 3:LOCATE 13,6:AM$="Amstrad M
agazine":PRINT AM$:PEN 1:LOCATE 15,10:PR
INT "1. JOGAR":LOCATE 15,13:PRINT "2. IN
STRUCOES":LO
CATE 15,16:PRINT "3. GRAVAR"
955 T$=INKEY$
957 LOCATE 13,6:PRINT AM$:PEN 3
960 IF T$="1" THEN GOTO 999
970 IF T$="2" THEN PEN 3:LOCATE 13,6:PRI
NT AM$:GOTO 800
980 IF T$="3" THEN PEN 3:LOCATE 13,6:PRI
NT AM$: SAVE "CRASH!"
985 LOCATE 13,6:PRINT AM$:PEN 1
990 GOTO 955
999 PEN 3:GOTO 100
```

CARACTERES EM TECNICOLOR

Como conseguir caracteres multicolores no ecrã, tanto em coordenadas LOCATE como em coordenadas PIXEL? Até agora tinhamos que o fazer em código-máquina, o que era pouco lógico uma vez que 99% dos programas funcionam suficientemente bem e com a rapidez necessária em BASIC.

SE não fosse através do código-máquina teríamos que construir vários caracteres que, sobrepostos, iríam formar as diferentes partes do caracter desejado e se iríam imprimir, com diferentes "canetas" e em modo transparente, um por cima do outro. Se, por exemplo, quisessemos um homem com o corpo azul, cabelo vermelho e a cara branca com um olho, necessitariamos de um caracter para o corpo, outro para o cabelo e outro para a cara, com um buraco para mostrar a cor de fundo através do olho; e tudo isto em modo 1! Se fosse no modo 0, poderiamos ter a necessidade de imprimir até 17 caracteres diferentes - o que se torna terrivelmente lento!

Sendo assim, para aqueles que queiram um método fácil e rápido de imprimir no ecrã caracteres multicolores (que, efectivamente são *sprites* muito pequenos) sem recorrer a grandes quantidades de código-máquina, desenvolvemos dois comandos RSX muito simples. Um deles é o |MPUT, para imprimir cada caracter, ou uma cadeia de caracteres multicolores, na última posição de impressão de texto. O outro é o |GPUT, para imprimir, tanto um caracter, como uma cadeia de caracteres, na posição do cursor de gráficos.

Para utilizar qualquer deles introduza simplesmente o comando, seguido de uma série de números separados por vírgulas. Por exemplo:

```
10 LOCATE 10,10: |MPUT,0,1,2,1
20 MOVE 100,100:|GPUT,4
30 MOVE: 200,40:|GPUT,5,6
40 LOCATE 20,3:|MPUT,7
```

Os números escritos a seguir aos comandos não se referem ao conjunto de caracteres normal mas sim a outro que será desenhado separadamente e sobre o qual falaremos mais à frente.

Como se pode ver, ambas as rotinas são muito simples de utilizar. Contudo, têm algumas limitações.

O comando |MPUT não coloca o cursor de texto por trás do último caracter que imprime; assim, se quisermos misturar caracteres multicolores e normais, temos que fazer um comando LOCATE antes de cada |MPUT. O comando |GPUT coloca o caracter na posição do byte mais próximo no ecrã, a qual não é sempre a posição do pixel visada. Isto significa que o cursor pode mover-se normalmente para cima e para baixo no ecrã, mas que só se moverá de 8 em 8 bits para a direita ou esquerda. Isto limita o movimento horizontal a meio caracter em modo 1 (quatro pixels) e um quarto de caracter em modo 0 (dois pixels). No modo 0 isto não constitui uma limitação excessiva mas se pensarmos que o movimento será muito desigual no modo 1, será melhor experimentar desenhar dois ou mais caracteres com a forma que quisermos ligeiramente modificada em cada um, de modo que, imprimindo-os um a seguir ao outro, se consiga um movimento suave. É esta a técnica utilizada no programa de demonstração.

PROGRAMAS

UM PREÇO A PAGAR

Tem que se pagar um preço em troca de todas estas facilidades: necessita-se de mais memória para cada caracter a imprimir. Em vez dos 8 bytes habituais, um caracter multicolor em modo 1 necessita de 16 bytes, e em modo 0 necessita de 32 bytes. Isto, no entanto, não é tão mau se considerarmos que teríamos que utilizar até 3 caracteres em modo 1 ou 15 em modo 0 para conseguir o mesmo resultado - processo que, aliás, seria muito mais lento. Um caracter em modo 2 continua a utilizar somente 8 bytes e a rotina funciona de facto com o modo 2, embora mantenha neste modo um pequeno problema: não pode ter mais que uma cor devendo utilizar-se em seu lugar o método normal de BASIC.

COMO FUNCIONA O PROGRAMA

Primeiro digite o programa 1, grave-o e execute-o. Se cometeu algum erro nas linhas de DATA, o programa vai indicar-lhe qual a linha em que este se encontra. Quando o programa funcionar sem erros, grave-o e re-inicialize o computador, desligando-o e ligando-o de seguida. Agora digite o programa 2 e grave-o. Este permitir-lhe-á não só desenhar os seus caracteres em todos os modos como ainda gravá-los em fita, ou disco.

O programa desenhador é muito simples e curto. Depois de lhe dizermos qual o modo gráfico que queremos utilizar, o programa entra nele. Para deslocar o cursor pelo caracter utilize as teclas de cursor e use a tecla [COPY] para activar e desactivar a "caneta". Para mudar a "caneta" carregue na tecla [P] (**P**EN) e introduza o novo número de "caneta". Para armazenar o caracter em memória carregue na tecla [S] e dê-lhe um número. Pode armazenar todos os caracteres que quiser enquanto toma nota da quantidade de memória que cada um gasta. Para re-editar um caracter introduzido anteriormente na memória carregue em [R] e introduza o número com o qual o armazenou. Quando acabar de desenhar caracteres carregue em [Q] (**Q**uit). Depois de o programa lhe ter perguntado se está seguro do que vai fazer, perguntar-lhe-à ainda, e por fim, quantos caracteres quer gravar. Lembre-se que o número de *sprites* começa em 0, e que o espaço que vai de 0 a 7 equivale a 8 *sprites*.

Agora, para ver o fruto do seu trabalho, deve re-inicializar a máquina e executar este programa

```
10 |MPUT "MODO ";MM:MD=1:IF
MM<>1 THEN MD=2
20 INPUT "QUANTOS SPRITES
";S:MODE MM
30 S=S*16*MD
40 H=HIMEM-1
50 MEMORY (H-S)
60 H=HIMEM+1
70 LOAD "SPRITES",H
```

que carregará os seus *sprites*.

Agora pode carregar e executar novamente o programa 1. Este moverá o HIMEM e vai auto-posicionar-se por cima activando-se de seguida. A partir deste momento já pode utilizar os comandos RSX previamente criados, e ver os seus *sprites* a aparecerem no ecrã. Não se esqueça que o comando

```
10 ' Programa 1
20 ' RSX para impressao em multicor
40 '
50 h=HIMEM-1:IF PEEK(0)=0 THEN 80
60 POKE 0,0:h=h-192:MEMORY(h-1)
70 lin=1000:enderesso=h:FOR t=1 TO 192 STEP 8
80 soma=0:FOR b=0 TO 7
90 READ a$:a$=UPPER$(a$):n=1
100 IF LEN(a$)<>2,THEN 370
110 b$=MID$(a$,n,1)
120 GOSUB 350:IF a=0 THEN 370
130 n=n+1:IF n=2 THEN 120
140 p=VAL("&"+a$):POKE enderesso,p
150 enderesso=enderesso-1:soma=soma+p
160 NEXT b:READ a
170 IF a<>soma THEN 370
180 PRINT "LINHA";lin;"OK":lin=lin+10
190 NEXT t
200 r=h+9:a=INT(r/256):b=r-(a*256)
210 POKE h+1,b:POKE h+2,a
220 r=h+17:a=INT(r/256):b=r-(a*256)
230 POKE h+4,b:POKE h+5,a
240 r=h+21:a=INT(r/256):b=r-(a*256)
250 POKE h+9,b:POKE h+10,a
260 r=h+30:a=INT(r/256):b=r-(a*256)
270 POKE h+41,b:POKE h+42,a
280 POKE h+63,b:POKE h+64,a
290 r=h+31:a=INT(r/256):b=r-(a*256)
300 POKE h+122,b:POKE h+123,a
310 r=h+194:a=INT(r/256):b=r-(a*256)
320 POKE h+31,b:POKE h+32,a
330 CALL h
340 END
350 a=INSTR("0123456789ABCDEF",b$)
360 RETURN
370 PRINT "ERRO NA LINHA";lin;"!":END
1000 DATA 01,29,4e,21,31,4e,c3,d1, 684
1010 DATA bc,35,4e,18,18,00,18,11, 408
1020 DATA 00,00,00,00,00,4d,50,55, 242
1030 DATA d4,47,50,55,d4,00,00,30, 708
1040 DATA 75,f5,af,18,03,f5,3e,01, 872
1050 DATA 32,3e,4e,f1,a7,c8,47,f5,1114
1060 DATA dd,23,dd,23,10,fa,cd,11,1000
1070 DATA bc,47,3e,02,90,f5,3a,3e, 832
1080 DATA 4e,a7,28,0a,cd,78,bb,2d, 852
1090 DATA 25,cd,1a,bc,18,1b,cd,c6, 910
1100 DATA bb,b7,cb,1c,cb,1d,cd,11,1055
1110 DATA bc,f1,f5,a7,28,08,47,b7,1143
1120 DATA cb,1a,cb,1b,10,f9;cd,1d, 958
1130 DATA bc,f1,c1,c5,f5,dd,2b,dd,1549
1140 DATA 2b,4f,06,03,80,47,79,e5, 680
1150 DATA ed,5b,3f,4e,dd,6e,00,26, 838
1160 DATA 00,29,10,fd,19,eb,e1,87, 930
1170 DATA 20,01,3c,47,c5,e5,06,08, 604
1180 DATA 3a,05,00,a7,28,09,fe,01, 534
1190 DATA 28,0b,fe,02,1a,18,08,1a, 391
1200 DATA a7,28,05,18,02,1a,ae,77, 557
1210 DATA cd,26,bc,13,10,e2,e1,cd,1122
1220 DATA 20,bc,c1,10,d7,f1,c1,10,1094
1230 DATA b2,c9,00,00,00,00,00,00, 379
```

|MPUT não moverá a posição do cursor, de modo que, se introduzir

```
|MPUT,0,0,0
```

a mensagem "Ready" irá sobreescrever os seus *sprites* a partir do momento em que acrescentar PRINT ao final do comando. E não se esqueça de situar o cursor de gráficos no ecrã com o comando MOVE para poder ver os caracteres, tal como em:

```
MOVE 100,100:|MPUT,0
```

Ora bem, agora deveríamos ter as rotinas a funcionar. Existem três métodos para imprimir caracteres. Podem imprimir-se os caracteres e apagá-los com |MPUT ou |GPUT sobrepondo-lhes um "caracter branco". Deve reservar um dos seus caracteres para fazer isto. Também pode utilizar-se o modo XOR para escrever os caracteres no ecrã, combinando-se estes com o que nele estiver no momento da impressão, da mesma forma como, com certeza já observaram, acontece em jogos como o "Manic Miner" e o "Sorcery", por exemplo. Utilizando este processo pode apagar-se o caracter impresso no ecrã imprimindo-se sobre ele o mesmo caracter. O terceiro método só imprimirá as partes do caracter que tenham alguma 'tinta'. Não imprime bytes zero. Suponhamos, por exemplo, que só queremos um caracter tamanho "modo1", mas em modo 0. Se neste modo só se desenha meio caracter (digamos, por exemplo, a metade esquerda) e se deixa o resto em branco, quando imprimirmos o caracter referido junto a outro obteremos normalmente o efeito bizarro de apagar os que estavam no ecrã, não aparecendo nada em vez disso. Este método de imprimir suprime este efeito e permite imprimir caracteres pequenos sem uma sombra à sua volta. Isto acaba por ser quase igual ao modo transparente normal. E significa que deve utilizar-se o primeiro método de impressão se quisermos apagar algo como um caracter em branco; de outro modo não acontecerá absolutamente nada.

Para seleccionar os métodos a utilizar execute simplesmente POKE 5,n onde n representa:

2 a 255 para impressão normal.
1 para impressão XOR.
0 para impressão transparente.

Para ver a rotina em acção digite a primeira parte do pequeno programa de demonstração que prepara os caracteres. De seguida, carregue e execute o programa 1 e depois carregue e execute a segunda parte do programa de demonstração para ver um balão a mover-se suavemente por todo o ecrã.

Como ponto final, o programa funcionará em qualquer lugar da memória, de maneira que se decidir desenhar um conjunto de caracteres diferente com o comando SYMBOL AFTER, faça-o antes de carregar e executar os caracteres multicolores e o programa 1 - assim tudo sairá bem.

```
10 'Programa 2 --------- DESENHADOR DE CARACTERES
20 INPUT "Modo";mm:md=1
30 IF mm<>1 THEN md=2
40 SPEED KEY 10,1:MODE mm
50 WINDOW#1,1,40/md,24,25
60 MEMORY 19999
70 q=1:BORDER 0
80 PEN 1:tt=TEST(x,398-y):LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(203)
90 k$=INKEY$:IF k$="" THEN 80
100 kk=ASC(k$)
110 LOCATE INT(x/(2*md))+10,INT(y/2)+1:PEN tt:PRINT CHR$(143):PEN q
120 x=x-(2*md)*((kk=&F3 AND x<(14*md))-(kk=&F2 AND x<>0))
130 y=y-2*((kk=&F1 AND y<14)-(kk=&F0 AND y<>0))
140 IF k$="p" OR k$="P" THEN PEN 1:INPUT#1,"Tinta";q:PEN q:CLS#1
150 IF kk=224 THEN te=1-(te=1):PEN 1:LOCATE 1,16:PRINT"OFF":IF te=1 THEN LOCATE
1,16:PRINT"ON "
160 LOCATE 1,15:PEN 1:PRINT "Tinta ";q;"  ":PEN q
170 IF k$="r" OR k$="R" THEN 240
180 IF k$="s" OR k$="S" THEN 220
190 IF k$="q" OR k$="Q" THEN 280
200 IF te=1 THEN LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(143):PLOT x,398-y
,q
210 GOTO 80
220 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
230 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE a+8*b+n,PEEK(c+(n*2048)+b):NEX
T n:NEXT b:GOTO 80
240 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
250 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE c+(n*2048)+b,PEEK(a+8*b+n):NEX
T n:NEXT b:GOSUB 260:GOTO 80
260 FOR b=0 TO 14 STEP 2:FOR n=0 TO 14*md STEP(md*2):tt=TEST (n,398-b):LOCATE IN
T(n/(2*md))+10,INT(b/2)+1:PEN tt:PRINT CHR$(143):NEX
T n:NEXT b:PEN 1
270 RETURN
280 PEN 1:INPUT#1,"Gravar caracteres";r$:CLS#1:r$=UPPER$(r$):IF LEFT$(r$,1)<>"S"
 THEN 200
290 INPUT#1,"Quantos";s:CLS#1
300 INPUT#1,"Com que nome";nome$:CLS#1:s=s*16*md:SAVE nome$,b,20000,s:GOTO 200
```

```
10 ;            M U L T I C O R
20 ;
30 ; Introduzir os comandos residentes |MPUT e |GPUT
40 ;
50           LD   BC,COMAND            ;aderesso da tabela de comandos
60           LD   HL,BYTES             ;aderesso de 4 bytes para o S.O
70           JP   #BCD1                ;
80 COMAND:   DEFW #A5CF                ;aderesso da tabela de nomes
90           JR   ENTRY2               ;salto para a rotina "MPUT"
100          NOP
110          JR   ENTRY1               ;salto para a rotina "GPUT"
120          DEFB #0
130 BYTES:   DEFB #0,#0,#0,#0          ;area para o uso do S.O.
140          DEFB "M","P","U",#D4      ;'MPU','T+#80'
150          DEFB "G","P","U",#D4      ;'GPU','T+#80'
160          DEFB #0                   ;marcador de fim da tabela
170 FLAG:    DEFB #0                   ;indica se se esta a processar MPUT ou
180 ;                                  ;GPUT
190 DIRDAT:  DEFW #A67C                ;aderesso dados caracteres
200 ;
210 ;aqui comeca a rotina GPUT
220 ;
230 ENTRY1:  PUSH AF                   ;preserva no. de caracteres
240          XOR  A                    ;apaga acumulador
250          JR   COMUM
260 ;
270 ;aqui comeca a rotina MPUT
280 ;
290 ENTRY2:  PUSH AF                   ;preserva no. de caracteres
300          LD   A,#1                 ;se e MPUT, flag=1 - se e GPUT
310 COMUM:   LD   (FLAG),A             ;flag=0
320          POP  AF                   ;recupera no. de caracteres
330          AND  A                    ;se nao ha caracteres retorna ao BASIC
340          RET  Z
350          LD   B,A                  ;se ha inicializa contador
360          PUSH AF                   ;grava no. de caracteres
370 CICLO1:  INC  IX                   ;e actualiza IX para que aponte
380          INC  IX                   ;para tras do primeiro caracter
390          DJNZ CICLO1
400          CALL #BC11                ;averigua o modo do ecran
410          LD   B,A                  ;fica 2-modo
420          LD   A,#2
430          SUB  B
440          PUSH AF                   ;preserva o resultado
450          LD   A,(FLAG)             ;carrega a flag (1: MPUT - 0: GPUT)
460          AND  A
470          JR   Z,GPUT2              ;se estamos em GPUT, salta
480          CALL #BB78                ;averigua posicao do CURSOR de texto
490          DEC  L                    ;converte as coordenadas logicas
500          DEC  H                    ;a fisicas e averigua o aderesso
510          CALL #BC1A                ;correspondente da memoria
520          JR   MPUT2                ;do ecran
530 GPUT2:   CALL #BBC6                ;averigua coordenados do CURSOR grafico

540          OR   A                    ;apaga o acarreio
550          RR   H                    ;converte coordenada y de "usuario"
560          RR   L                    ;a coordenada y de "base"
570          CALL #BC11                ;averigua o modo do ecran
580          POP  AF                   ;recupera 2-modo
590          PUSH AF                   ;e torna a preserva-lo
600          AND  A                    ;testa se e 0 (modo 2)
610          JR   Z,PIXEL              ;se e modo 2, salta
620          LD   B,A                  ;se nao, passa 2-modo a B
630 CICLO2:  OR   A                    ;apaga o acarreio
640          RR   D                    ;converte coordenada X de "úsuario" a
650          RR   E                    ;coordenada X de "base"
660          DJNZ CICLO2
670 PIXEL:   CALL #BC1D                ;calcula aderesso de pixel no ecran
680 MPUT2:   POP  AF                   ;recupera 2-modo
690          POP  BC                   ;recupera no. de caracteres
700 OUTRO:   PUSH BC                   ;grava no. de caracteres
710          PUSH AF                   ;grava 2-modo
```

```
 720            DEC  IX                    ;aponta o caracter
 730            DEC  IX
 740            LD   C,A                   ;carrega en C 2-modo
 750            LD   B,#3
 760            ADD  A,B                   ;soma-lhe 3
 770            LD   B,A                   ;passa 2-modo+3 a B
 780            LD   A,C                   ;recupera 2-modo
 790            PUSH HL                    ;preserva aderesso do pixel
 800            LD   DE,(DIRDAT)           ;DE=#a67c principio dos dados
 810            LD   L,(IX+0)              ;de caracter
 820            LD   H,#0
 830 CICLO3:    ADD  HL,HL                 ;calcula aderesso em que se encontra
 840            DJNZ CICLO3
 850            ADD  HL,DE
 860            EX   DE,HL                 ;passa aderesso a DE
 870            POP  HL                    ;recupera aderesso do pixel
 880            ADD  A,A                   ;duplica 2-modo para calcular a largura
 890            JR   NZ,LARG               ;do caracter.Se e modo 2,largura=1
 900 LARG:      INC  A                     ;largura de caracter a B
 910 COLUMN:    PUSH BC                    ;preserva largura e 2-modo
 920            PUSH HL                    ;preserva aderesso do pixel
 930            LD   B,#8                  ;inicializa contador
 940 CICLO4:    LD   A,(#5)                ;carrega o mododo impressao
 950            AND  A                     ;prova se e transparente
 960            JR   Z,TRANSP
 970            CP   #1                    ;prova se e um (XOR)
 980            JR   Z,MODXOR              ;se nao,e modo normal
 990            CP   #2
1000            LD   A,(DE)
1010            JR   NORMAL
1020 TRANSP:    LD   A,(DE)                ;carrega primeiro byte do caracter
1030            AND  A                     ;se e zero, passa a NORM2
1040            JR   Z,NORM2
1050            JR   NORMAL                ;e se nao a NORMAL
1060 MODXOR:    LD   A,(DE)                ;carrega byte do caracter
1070            XOR  (HL)                  ;XOR com o do ecran
1080 NORMAL:    LD   (HL),A                ;envia byte para o ecran
1090 NORM2:     CALL #BC26                 ;calcula aderesso do byte inferior do
1100 ;                                     ;ecran
1110            INC  DE                    ;aponta para o seguinte byte do caracter
1120            DJNZ CICLO4                ;repete para os oito bytes
1130            POP  HL                    ;recupera aderesso da esquina superior
1140 ;                                     ;do caracter.
1150            CALL #BC20                 ;calcula aderesso da coluna seguinte
1160            POP  BC                    ;recupera altura do caracter
1170            DJNZ COLUMN                ;repete ate completar o caracter
1180            POP  AF                    ;recupera 2-modo
1190            POP  BC                    ;recupera no. de caracteres
1200            DJNZ OUTRO                 ;repete para todos os caracteres
1210            RET                        ;volta ao BASIC
```

CORREIO DOS LEITORES

- Sendo os computadores sinclair/spectrum uma sub-marca da Amstrad, porque é que vocês não fazem nunca referência a qualquer assunto relacionado com esses computadores ?!
- Também gostava de sugerir que na vossa revista, criassem uma secção, ou pelo menos escrevessem artigos sobre o sistema operativo CP/M Plus.

D. Coelho

AM - Quem sabe se num futuro próximo os utilizadores do Spectrum não vão ter uma surpresa com uma outra publicação?
- Lá chegaremos aos artigos de CP/M. Fique com a AM mais algum tempo.

Sugiro que também haja programas em disquetes de 3.5" e se há que nos diga quais. Por acaso não tenho um Amstrad mas há muitas pessoas que compram a AM e não têm um Amstrad. Por isso que nos digam em que linguagem é que os programas são feitos, para darem noutros computadores.

AM - Todos os programas de Free-Soft estão já disponiveis em formato 3.5", e com o tempo teremos todos os programas em todos os formatos. Para já estamos a estudar a situação, uma vez que é um pouco dificil comercializar um "Flight Simulator" da Microsoft se antes tivermos aberto a package para transferir o programa para 3.5" (esta, por exemplo, é uma das packages que é fornecida de origem apenas em formato 5.25"). Em relação à sua segunda sugestão, queremos reafirmar que tentamos sempre segui-la. Cada palavra ou listagem dedicados aos PC´s Amstrad nesta publicação aplica-se, pelo menos noventa e tal por cento das vezes a todos os outros PC´s compativeis do mercado. As matérias ligadas aos CPC´s e PCW´s já possuiem um caracter mais especifico, mas no fundo, quando alguém adquire um PC compativel procura, para além de outras coisas, possuir uma máquina idêntica a muitas outras, entrando num standard, com todas as vantagens e desvantagens que daí lhe possam surgir.

1 - Queria saber se o produto posto à venda no "CLUBE AM", nomeadamente o produto ´Drives de 5.25"´ Ref. 903, traz as respectivas instruções de instalação ?
2 - No concurso Publinfor/Socartel, qual é a morada para onde devem ser remetidos os trabalhos?
3 - Porque será que a revista "AMSTRAD" chega à cidade do Funchal só no final do mês. A revista nº9 de Fevereiro só chegou no dia 27 desse mês ?

AM - 1. Sem conhecermos o computador onde pretende instalar a drive de 5.25" não poderemos enviar-lhe instruções de montagem da referida drive, já que tal tarefa varia em alguns detalhes de computador para computador (apesar de ser sempre bastante simples). Assim, não tinha sentido enviar as instruções para montar a drive num PC Amstrad quando você pretende montar a drive num PC Philips, por exemplo. Por outro lado não seria viável para nós fornecermos a drive com instruções de montagem adequadas a todos os PC´s do mercado. Por tudo isto apenas podemos colocar à disposição dos leitores toda a nossa ajuda

telefónica ou escrita (com preferência para esta última hípotese), afim de concluir da melhor maneira possivel esta ou qualquer outra tarefa, quer no âmbito do hardware, quer no âmbito do software.
2. A morada para onde devem ser enviados os trabalhos para os prémios mensais PUBLINFOR/SOCARTEL é a seguinte:
PUBLINFOR/SOCARTEL
Centro de Escritórios das Laranjeiras
Urbanização das Laranjeiras
Praça Nuno Rodrigues dos Santos, Nº7, 2 Piso, sala 13
1600 LISBOA
Quaisquer dúvidas sobre este concurso podem ainda ser esclarecidas através do telefone 7269011.

CORREIO DOS LEITORES

Envios de pequenos textos (artigos, "dicas", etc.) podem ainda ser efectuados através do Fax 7269985.
3. Tanto os atrasos que afectavam a saída da AM para a rua como os que se verificavam na nossa secção de encomendas, foram até há muito pouco tempo os nossos maiores "problemas crónicos", neste momento tanto um como o outro estão resolvidos, a prova é esta AM na rua NA ÚLTIMA SEXTA-FEIRA DO MÊS, e as encomendas a serem despachadas com a rapidez desejada nestes casos.

A minha sugestão é que a AM publique anualmente uma capa, de modo a que as revistas da AM possam ser guardadas de uma forma eficiente.

Pedro

AM - Internamente a ideia já tinha surgido e as capas estão por aí a "estoirar" com caracteristicas muito interessantes: não vai necessitar de mandar encadernar a sua colecção de AM, e a qualidade da capa estará de acordo com a qualidade que tentamos sempre deixar em cada número desta publicação.
Em todo o caso, obrigado pela sugestão, e mantenha-se atento.

Para começar, adoro a AM, no meu entender o que falta é programas para PCW 9512, e noções para o Malard BASIC 80.
Tenho 2 PCW 9512 e 2 PCW 8512 apenas como processadores de texto, gostaria de aproveitar melhor as suas capacidades.

Fabio Resende - Vila do Conde

AM - Continue com a AM e vai ver que não se arrepende. Não podemos agradar a gregos e troianos ao mesmo tempo. Até agora temos agradado mais aos gregos, mas os troinanos nunca foram, e não serão, esquecidos na "Revista dos Utilizadores Amstrad".

Sou o Ricardo e compro todos os meses a vossa revista [nota do ed.: a revista não é nossa. A revista é NOSSA, e como tal é tanto sua como nossa]. Na revista de Fevereiro de 1989 fazem referência a um concurso onde se pode ganhar 20 mil escudos, mas não fazem referência á maneira como se mandam os programas.
Tenho ainda uma pequena critica a fazer em relação à rubrica "COMPRO/VENDO/TROCO", pois tem poucas quadrículas (não se consegue escrever um anúncio em boas condições).

AM - Em relação à primeira questão, e para não ocupar espaço desnecessáriamente, veja uma resposta dada a outro leitor nesta mesma revista.
No que diz respeito à secção "Compro/Vendo/Troco" temos recebido muitas críticas, especialmente porque alguns leitores não enviam o contacto correcto para se efectuar a "transacção" (e outros não enviam qualquer contacto), e nós continuamos a publicar os anúncios.
Se soubessemos que ao acrescentarmos mais quadrículas no postal este problema ficava resolvido, não tenha quaisquer dúvidas que o faríamos. A situação, contudo, está longe de ser essa. Num dos próximos números é possivel que se encontre uma solução para todos estes problemas.

1- Porque será que é o postal 3 que é RSF, quando deve ser acompanhado do cheque para pagamento? Logo não serve...
2- Há anúncios publicados sem morada ou número de telefone. Para que servem assim ?
3- Para quando a possibilidade dos pagamentos através do cartão UNIBANCO ?
4- Como "continuamos em português", atenção aos "." em vez das ","! e os estrangeirismos entre aspas ou em cursivo! CONTINUEM !

João L. Santos

AM - 1. A revista Nº9 constitui uma excepção à regra porque foi alvo de uma troca de frentes com versos nas folhas dos postais. Veja as números 8 e 10 e confirme o facto.
2. Veja resposta ao leitor anterior.
3. Fica a sugestão.
4. Se Camões tivesse escrito um longo texto sem qualquer tipo de pontuação todos iríamos tentar

arranjar meia dúzia de palavras menos vulgares para definir esse tipo de discurso, infelizmente não temos Camões como colaborador da revista, e como tal todas as trocas de pontos por virgulas, e vice-versa, estão condenados desde o inicio. Curiosamente embora não tenhamos "o poeta" como colaborador temos um Fernando, pessoa que por vezes também comete alguns erros sem se preocupar muito com isso visto que não dirige uma revista de literatura.
Sobre as traduções apenas lhe fazemos um único pedido. Traduza-nos as seguintes três palavras de modo a que os outros leitores saibam a que se refere quando as aplica:
a) BUFFER
b) DRIVE
c) SOFTWARE

O único mal desta revista é não publicarem programas para o PC (GW-BASIC), e publicarem doses demasiadas de programas para o CPC.

AM - A maior "fatia" da Amstrad Magazine será sempre dedicada aos PC´s compativeis IBM. As razões que nos levam a agir desta forma explicam-se pelo facto de a evolução no mundo dos utilizadores tender cada vez mais para o standard dos computadores pessoais. Logo, um utilizador de um CPC, ou de um PCW, hoje, é um potencial utilizador de PC´s amanhã.
Aquilo que não podemos fazer é esquecer pura e simplesmente os leitores que possuem os CPC´s e PCW´s. Nós continuamos a viver no "hoje", e esta é a revista de TODOS os utilizadores Amstrad (já hoje disse isto não sei onde).

Sou assinante da Amstrad, e gostaria de ver publicado na revista mais programas e "truques" para os PC´s, dando mais espaço para estes e para os da série 2000 (2086).
Parabéns pelo vosso trabalho. Continuem.

Amâncio Manuel - Porto

Já que comercializam, por exemplo, disquetes, porque não também:
- fitas para as impressoras Amstrad
- papel
- outros acessórios como, por exemplo, caixas para disquetes, joysticks, etc.
- jogos Amstrad
- PC add-ons e modems Amstrad.

AM - A pouco e pouco tudo isso estará disponivel no clube AM. Já viu as novidades deste mês.

- Criação de um espaço de apresentação e divulgação, das várias linguagens dos computadores (bem como as suas principais aplicações, etc.)
- Mais e melhor divulgação do software de aplicação para PC´s (2086) - principalmente relacionados com electrónica e circuitos impressos.
- Porque não fazer pequenos cursos simples e objectivos, das linguagens de computador existentes, ou das mais "populares" e usadas.
- Distribuição a tempo e horas
- Mais programas de FREE-SOFT por cada disquete de 3.5"
- Continuem.

Amâncio M. - Porto

AM - A data de distribuição parece estar de acordo com o que sempre anunciámos pela primeira vez na história da AM. O resto passará por aqui, a pouco e pouco. Leia, por exemplo, a série de artigos que iniciamos neste número dedicada a bases de dados.

+ Informações sobre PPC
+ programas compativeis IBM
+ Análises de programas compativeis DOS
informações sobre ligações à rede telefónica.
+ Informação sobre a criação de adaptações a Portugal de software estrangeiro (codepages, characters sets, etc.)
Drivers para software a usar em Português.

AM - O WINDOWS NÃO TINHA... UM TECLADO PORTUGUÊS

- Sugiro a criação de um dicionário de termos informáticos, tanto de Soft como de hardware.
- Criação de postais de divulgação de assuntos informáticos (soft e hardware com as respec-

tivas caracteristicas, aplicações, e possibilidades de expansão, etc.)
- Mais e melhor divulgação dos computadores que apresentam (caracteristicas, etc., etc....)
- Continuem.

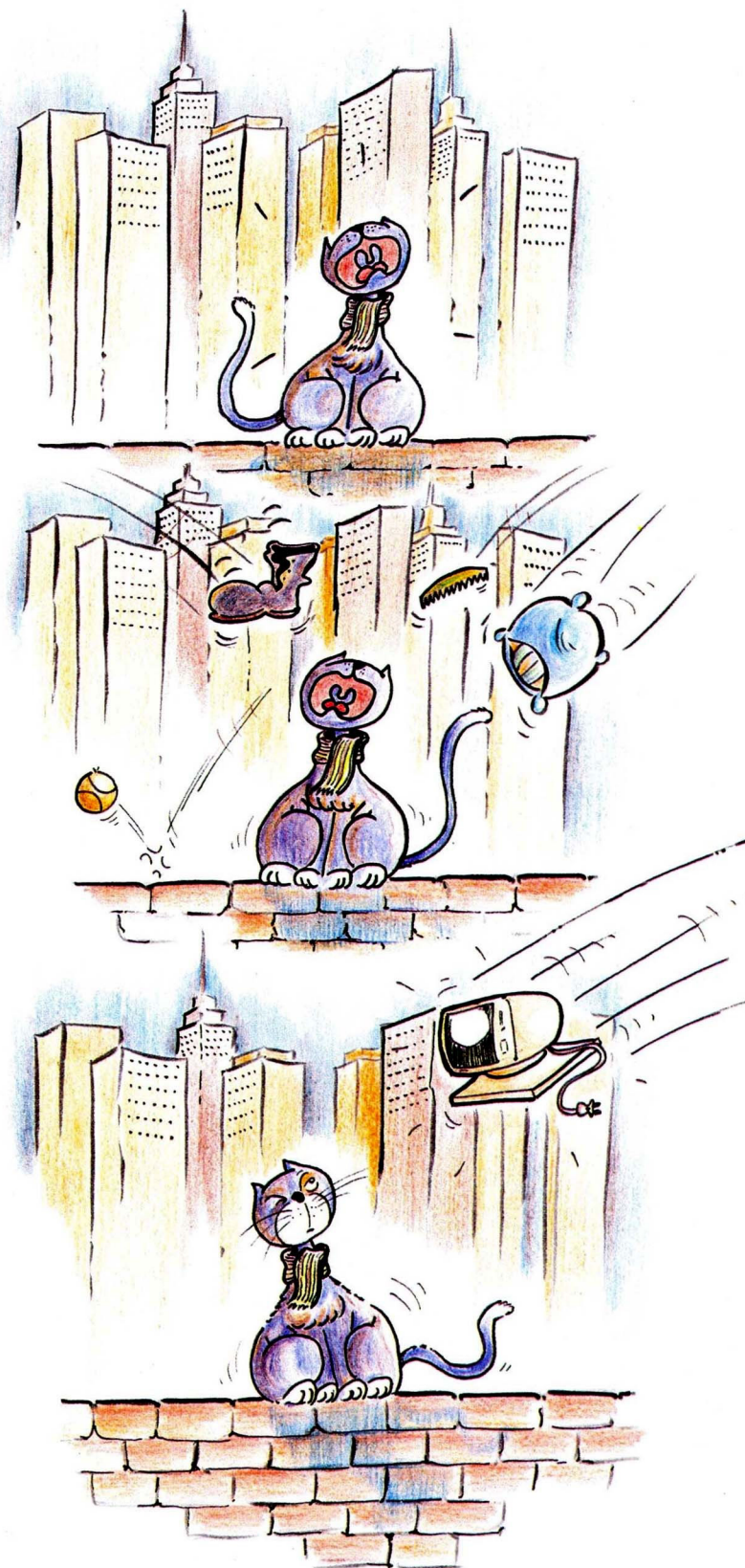
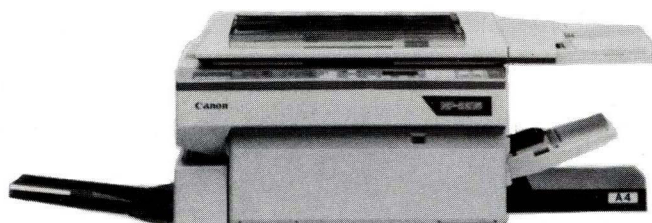
Amâncio - Porto

AM - A forma como o PC 2386 foi analisado nesta AM (e como serão analisadas todas as máquinas de futuro), foi uma consequência parcial deste postal. Continuem todos a mandar mais sugestões.

Devia haver uma lista para os jogos CPC 464 como há para os PC 1512 e 1640, e mais utensilios para os CPC 464, como interfaces joystick, e programas educativos.

AM - Veja o CLUBE AM, neste mesmo número.

Eis alguns erros grosseiros de português na vossa edição de Fevereiro:
** - "... tomar forma à (?) cerca..." (editorial, pág. 1)*
** - "... concurressem" (2 vezes) - pág. 3*
** - "... convencido que (?)..." - pág. 6 - col. 2 - lin. 30*
** - "... quanto ao caso... permitem (?)..." - pág. 12 - col. 2*
** - "... que providência (?) utilitários..." - pág. 14 - col. 3*
** - "... que providência um jogo..." - idem*
** - "... ao qual estão ligadas uma série..." - idem*
** - "... dicidiram..." - pág. 21 - col. 1 - lin. 36*
** - "... ciêntificas... ciêntificos... (?)" - pág 22 - col. 1*
** - "... direcçional...(?)" - pág. 23 - col. 2 - penult. linha*
** - "... concurrentes... (?)" - 2 vezes - pág. 36 - col.3*
** - "... re-afirmarmos...(?)" - pág. 38*
** - "... concerteza... (?)" - pág. 46*
** - "... caractéres...(?)" - 2 vezes - pág. 61*
etc., etc. ... etc... etc. ...
Nas pág. de publicidade não encontrei uma única falha. As agências sabem o que fazem...
Nas inúmeras palavras ou expressões em inglês vocês nunca falham! Ora vamos lá a ter um pouco mais de respeito pela nossa língua! Haja decoro!

AM - se não fossem todos os seres perfeitos deste mundo o que seria de nós, péssoas estúpidas, que cometemos muitos erros, muitas vezes.
Obrigado por existir. Obrigado por nos ter feito criticas tão pertinentes. Muito obrigado por tudo, a quem quer que seja.
P.S. : Não se esqueça de ler no próximo número da AM a análise temática da "Peregrinação", de Fernão Mendes Pinto.

COMPRO

Impressora de 80 colunas configurada em série ou paralelo. Escrever para (indicando preço) Apartado 284560 Penafiel.

Software que funciona com o sistema operativo CP/M Plus. Também compro impressora de 80 colunas. Apartado 284560 Penafiel.

Troco todo o tipo de software em 3.5" e 5.25" para PC's sonosoft. Rua Rocha Peixoto, 16 r/c 4490 Póvoa de Varzim.

VENDO

Impressora Seikosha GP-5 S como nova para ZX Spectrum + 3 rolos, exemplos e manual por 19 cts. Contactar (043) 52636 Almeirim.

Amstrad CPC 664 + monitor GT65 + impressora DMP1, c/ manuais e embalagens. Pouco uso, óptimo estado. 70 cts. Tel. (01) 8145634.

Computador Spectrum +3 + joystick + gravador + 2 disquetes + fio para gravador + 35 jogos tudo com pouco uso. Tel. 2904152.

Vendo Timex 2048 + monitor + gravador + joystick + 550 jogos por 55 cts. Tudo em bom estado. Tel. 2536039 Corroios.

Compact Disc (duplo) dos Pink Floyd "Umagumma" por 3900$00 contactar Bruno Barão, Travessa Pedro Samora, nº4 - 8200 Albufeira.

Your Spectrum ano 1987, 12 revistas, preço 2500$00; troco soft-PC + de 80 programas, contacte António Rodrigues, Tel. (02) 682371 Porto.

Timex 1500/16K c/ PG.S útil. Timex 2048, impressora 2040 bons preços. Contacte Alcino, R. Padre Américo 200 V. Andorinho 4400 Gaia. Tel. 7820768.

ZX Spectrum +2 com monitor e joystick Turbo II + 30 jogos, só por 40 cts. Mário Manuel, Caldas de S. Jorje Tel. (056) 97303.

Vendo: 5.25", Sk-compress/Copy II PC/ PCTools/QDOS/Norton; 31/2: Lotus/DBase III/GW-Basic. Tel. (noite) 4365891, disquetes incluídas no custo.

Vendo Spectrum 48K em bom estado com manuais + 100 jogos conhecidos + cabos p/ gravador + transformador corrente 15 cts. Tel. 4106315.

Apple II/C + monitor + SUP. Monitor + software entre o qual Apple Works = B. Dad. Proc. F. Cal. Manuais c/ novo 100 cts. Tel. (02) 9481295 das 19-21 horas.

Spetrum Plus + gravador digital tudco em bom estado por 20 cts. Contactar António. Tel. 9812234.

Micro computador Casio FX-750 P programável em Basic com, 4Kb memória expansível a 16Kb, preço 16 cts. Contactar Paulo. Tel. 4312038.

PC-Tools Deluxe-Turbo BasicSsidekick-Copywrite-Graphics Assistan-PC format/10 cts. Contactar (052) 53250.

Vendo: 5.25", Clipper/SK/UP/PCTools/Norton/Edit/Copywrite/Explorer/Popalarm/QDos/TD. Tel. (noite) 4365891 disquetes incluídas no custo.

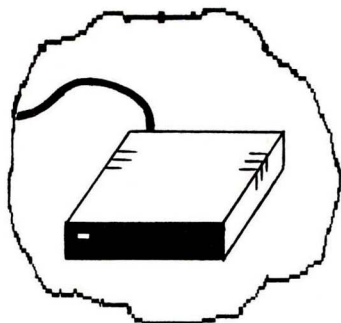
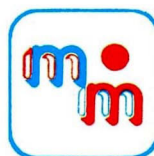
Philips MSXVG. 8020 e monitor BM:7552 em bom estado por 50 cts. Faço desconto, oferta joystick + 15 jogos. Cabeço 3125 Louriçal, tel. (036) 95418.

ZX Spectrum + 128K com monitor Neptun 156 mais gravador e cerca de 70 jogos tudo novo por 47 cts. Tel. 52467 Albufeira.

Vendo Amstrad PC 1512 - 1640 em segunda mão com garantias preços 30% a 40% mais abaixo que os novos. Carlos Miguel Tel. 9812312 qualquer hora.

CPC 464 + DDI Disk drive livros + revistas + caixa disquetes + Pascal + jogos + mini office. Tel. 2043336.

Vendo Amstrad PC 1640 duas drives, monitor monocromático e impresora Epson LX-800 em conjunto ou separadamente. Tel. 730662.

Vendo ZX Spectrum 48K e jogos para ZX Spectrum. Tel. 730662.

Programas para PC's compatíveis IBM c/ manuais incluindo jogos. Contactar João Silva - Apartado 21224 - 1131 Lisboa Codex.

DEC Raimbow DD+ Impr. La-50, monitor Ambar + cabos + DBase III + Multiplan + Logo + Pascal + MBasic, ambientes DOS e CP/M Tel. 768462 (Lisboa).

Vendo e troco software em disquetes de 5.25" ou 3.5". Crio software de aplicação manual. Meneses. Rua Rocha Peixoto 16 r/c 4480 Póvoa.

Geoceu-Grupo Ovnilog. Ciências Esot. Universais/ estudo de personalidades: envie 500$00 com nome e data de nascimento para Apt. 4278-4004 Porto.

Software em disquetes para Amstrad CPC, mais de 250 programas, contactar João Dinis, Rua Júlio Dinis, nº91 1º Esq. - 4000 Porto.

Impressora Seikosha SP-180 por 39 cts. Monitor Neptun por 7.500$00. Computador TC 2068 por 16 cts.

ZX Spectrum +3. Reduzida utilização, como novo por 50 cts. Escrever para António Augusto Silva. Rua Alexandre Herculano 173 - 2º Esq. 3880 Ovar.

TC68 + Interface + Joystick + Leitor + 60 programas (recentes) + curso Basic por 25 cts. Gabriel Fontes Tavares, Casal do Monte Fiaes. 4535 F. Norte.

Amstrad PC 1512 mono 2 drives c/ software, Tel. (056) 61492 Oliveira de Azemeis.

Monitor Zenith, Impressora Seikosha, Sinclair QL, estabilizador, livros e cassetes suplementares por bom preço. Alexandre 053-621706.

Amstrad CPC 664 + monitor GT65 + impressora DMP1, c/ manuais e embalagens. Pouco uso, óptimo estado por 70 cts. Tel. 01-8145634.

Vendo 48K+ com monitor Philips Fósforo verde com joystick e os respectivos Interface. Contactar Tel. 9886866 bom preço.

TROCO

Troco software - PC's bons programas/jogos. Contactar Nuno Jorge Gabriel Rodrigues. Rua Picadeiro, 50 - 5300 Brangança. Tel. 22339.

Troco software para PC's. Procuro o jogo Dr. Livingstone, suponho. Pedro Pinheiro Corujeira - S. Martinho do Bispo 3000 Coimbra. Tel. 814397.

Troco software para PC vindo do Brasil. Contactar Fernando F. Oliveira, Rua da Tapada, 35 Canas de Senhorim 3525.

Geoceu-grupo ovni/ciências esotéricas estuda aplicações/ pesquisa computorizada sobre tema. Contacta-nos APT 4278-CX 4004 Porto.

Troco e vendo software PC compatível (preço só da disquete) contacte António Rodrigues. Tel. 02-682371.

Software para PC 1640 envie a sua lista e receberá a minha. Carlos Santana, Rua Carlos Pereira, 2 c/v Dta. 1500 Lisboa.

Inforangelo-PC software - troca de soft. para PC's compatíveis, mande lista. Inforangelo-PC Soft. Ap. 117 3080 Fig. da Foz.

Troco ou vendo programas e jogos. Tenho ainda um programa de manipulação de ficheiros. Ricardo Pinto, Travessa Fernandes Tomás, Ovar.

Troco software PC 1512/PCW 9512 toda a gama de gestão/ CAD/Jogos, etc. Fabio Resende Tel. 620222 E. C. Premar-51 - 4490 Póvoa de Varzim.

Software para PC's, envia a tua lista para Joaquim dos Santos Soares - C. C. nº 450. 4590 Paços de Ferreira, encontrarás amigo.

Jogos software para PC. Tel. 7151393 João Paulo.

Modula-2 gostaria de trocar programas, ideias e informações com utilizadores desta linguagem. Contactar Fernando 02-930424.

Troco software para PC's com-

COMPRO/VENDO/TROCO

patíveis e mais de 200 programas. Ricardo Pereira, Rua das Rosas, 36 r/c Esq. Moreira Maia, 4470 Maia.

Ideias e software para Amstrad PC e compatíveis, contactar Bruno Barão, Travessa Pedro Samora, nº 4 - 8200 Albufeira.

Preciso informações do manual Clipper (comp. DBase 3+) dou soft PC compatível António Rodrigues. Tel. 02-682371.

Software utilitário para Spectrum +3 e compro software que trabalhe sob CP/M Plus. Escrever para apartado 28 4560 Penafiel.

Hoje, existe o novo Amstrad PCW 9512.

**Mais do que uma evolução natural na escrita, o novo AMSTRAD PCW 9512 é uma revolução.
Corrige, alinha frases, substitui palavras, personaliza cartas, a impressora faz cópias automaticamente... e, todos os textos ficam registados em arquivo numa diskette, prontos a serem utilizados.
O novo processador de texto AMSTRAD PCW 9512 executa todas estas tarefas com rapidez, simplicidade e eficiência. É mesmo revolucionário!
Não pense que o AMSTRAD PCW 9512 lhe vai custar mais do que uma máquina de escrever electrónica, lembre-se que AMSTRAD é qualidade a baixo preço. Ponha no lixo a sua incompetente e ultrapassada máquina de escrever.**

Amstrad PCW 9512
A revolução na escrita.

Cominfor
COMPANHIA PORTUGUESA DE INFORMÁTICA

AMSTRAD